Velociraptor

Digging Deeper!

Velociraptor Installation and overview

Introducing the little green reptile!

Module overview

In this module we introduce the tool and explain the rationale behind its design.

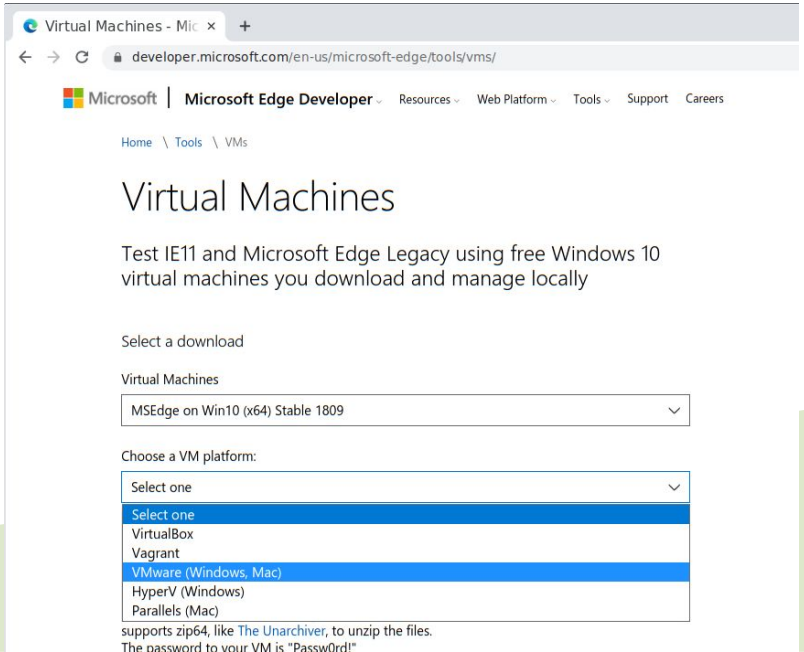
We will deploy Velociraptor in a cloud environment - We aim to be as close to how one would deploy it on a real deployment as possible.

We will play with the GUI and introduce some of the main concepts

Prerequisites

In order to follow along with this workshop you will need to use a windows VM with administrator level access.

You can grab a free VM from [Microsoft]



Prerequisites

In the first part of this session I will be installing Velociraptor in the cloud environment on GCP.

Because of the size of this course we can not provide people with the cloud infrastructure so you will need to do this on your own cloud account later - this is a demonstration only.

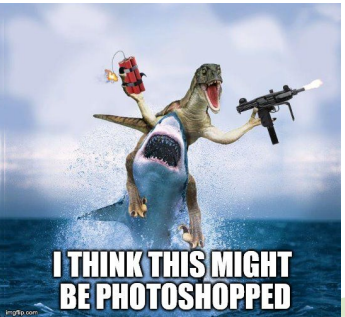
There are a lot of screenshots in the slides to allow you to replicate this later with your setup - we recommend practicing this a couple of times.

What is Velociraptor?

Velociraptor is a unique DFIR tool, giving *you* power and flexibility through the Velociraptor Query Language (VQL)

VQL is used for everything:

- ❏ Collecting information from endpoints (also called *clients*)
- ❏ Controlling monitoring and response on endpoints
- ❏ Controlling and managing the Velociraptor server.



History

Velociraptor draws inspiration from two major projects:

- GRR https://github.com/google/grr
- OSQuery https://github.com/osquery/osquery

Velociraptor vs GRR

In common

- ❏ Hunting across large number of endpoints
- ❏ Can collect file data
- ❏ Free Open source (FOSS)

Different

- ❏ Much faster
- ❏ Lower footprint
- ❏ A flexible query language
- ❏ Very simple to deploy
- ❏ Event based queries
- ❏ Commercially supported FOSS

Velociraptor vs OSQuery

In common

- ❏ Rely on a query language to access machine state
- ❏ Single binary with no dependencies
- ❏ Multi-platform

Different

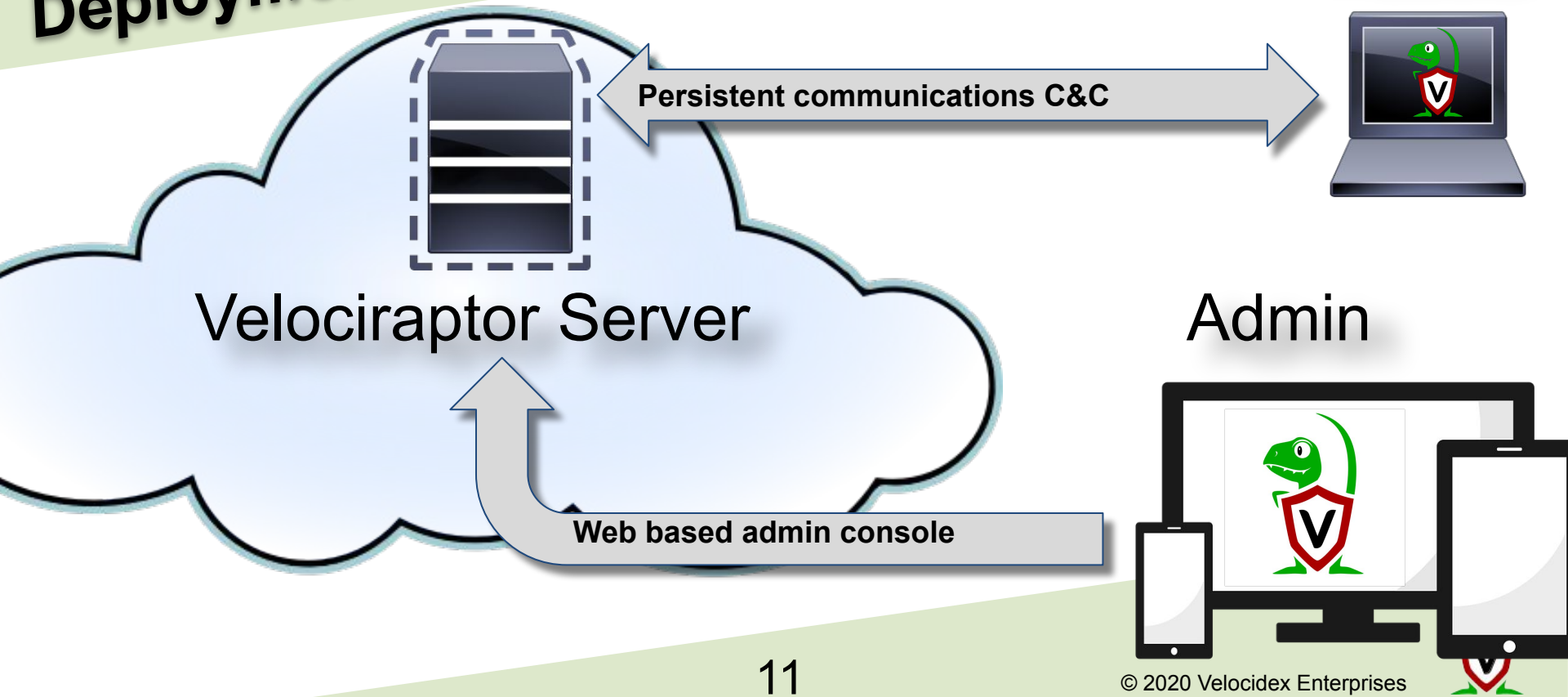
- ❏ VQL is much more powerful and intuitive than SQL
- ❏ Much faster than OSQuery
- ❏ Can transfer file data
- ❏ Can modify the system
- ❏ Remote client/server control and orchestration in the same tool.

Velociraptor vs OSQuery

- ❑ F erful
  t
- ❑ S
  c
- ❑ M ntrol
  same

rises

Deployment overview



Typical deployments

Velociraptor is very efficient and scalable:

- Server simply collects the results of queries - clients do all the heavy lifting.
  - Client memory and CPU usage is controlled via throttling and active cancellations.
- Server is optimized for speed and scalability
  - Concurrency control ensures stability
  - Bandwidth limits ensure network stability

Typical deployments

Current recommendations

- 10k-15k clients - single server with file based data store (usually cloud VM).
  - SSL load is the biggest load - TLS offloading helps a lot!
  - 8 GB RAM/8 cores is generous towards the top of the range.
- We recommend Ubuntu/Debian server

Multi-Frontend configuration

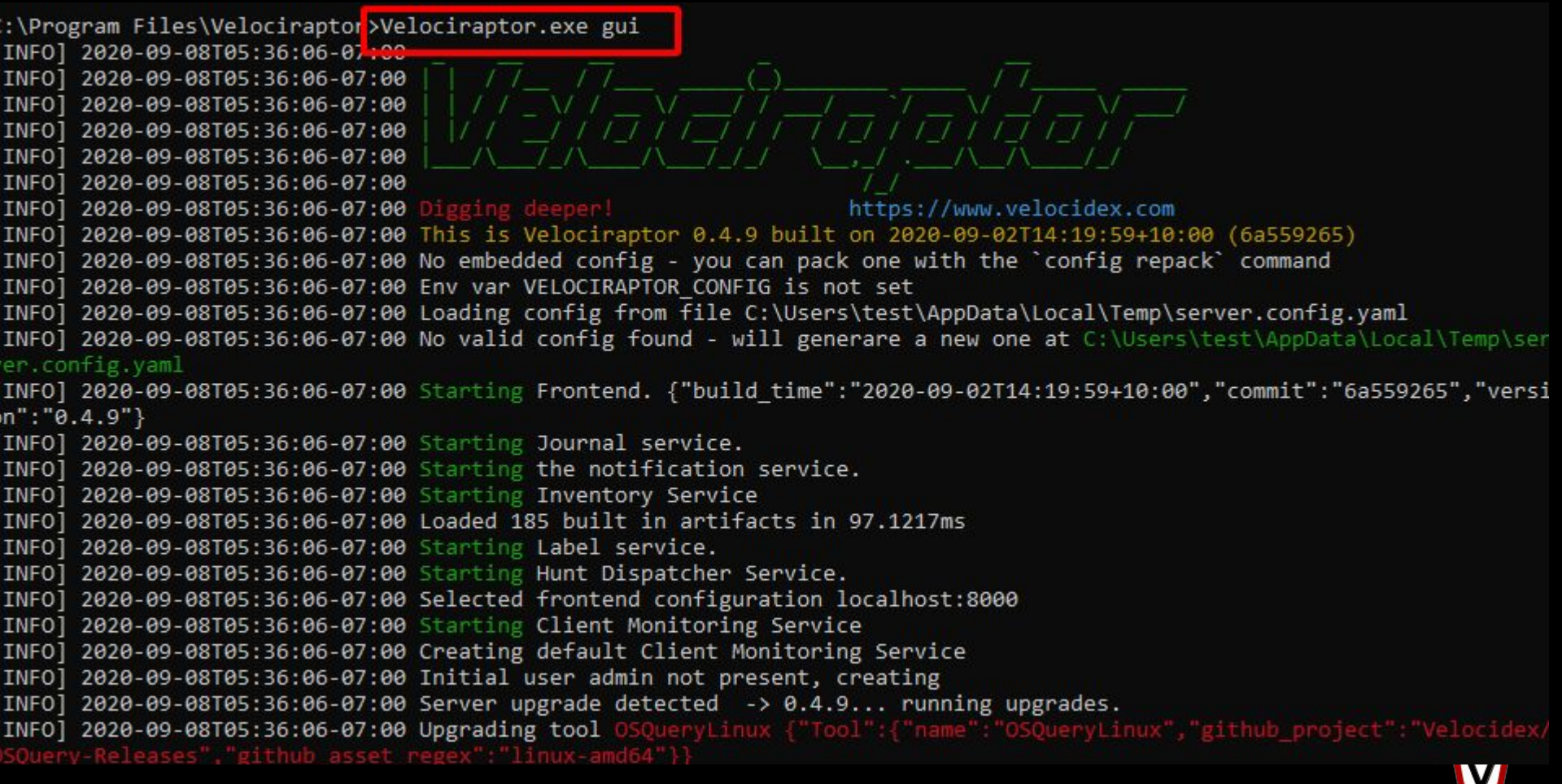
- Available since 0.5.9 - suitable for > 10k endpoints
- Still considered experimental - help us test it!
- Master/Minion model
- Outside the scope of this course but you can find more information in our blog post

Deploying Velociraptor

Run Velociraptor on your machine

Download Velociraptor from GitHub (.msi or .exe)

"C:\program files\Velociraptor\Velociraptor.exe" gui



Self Signed SSL mode

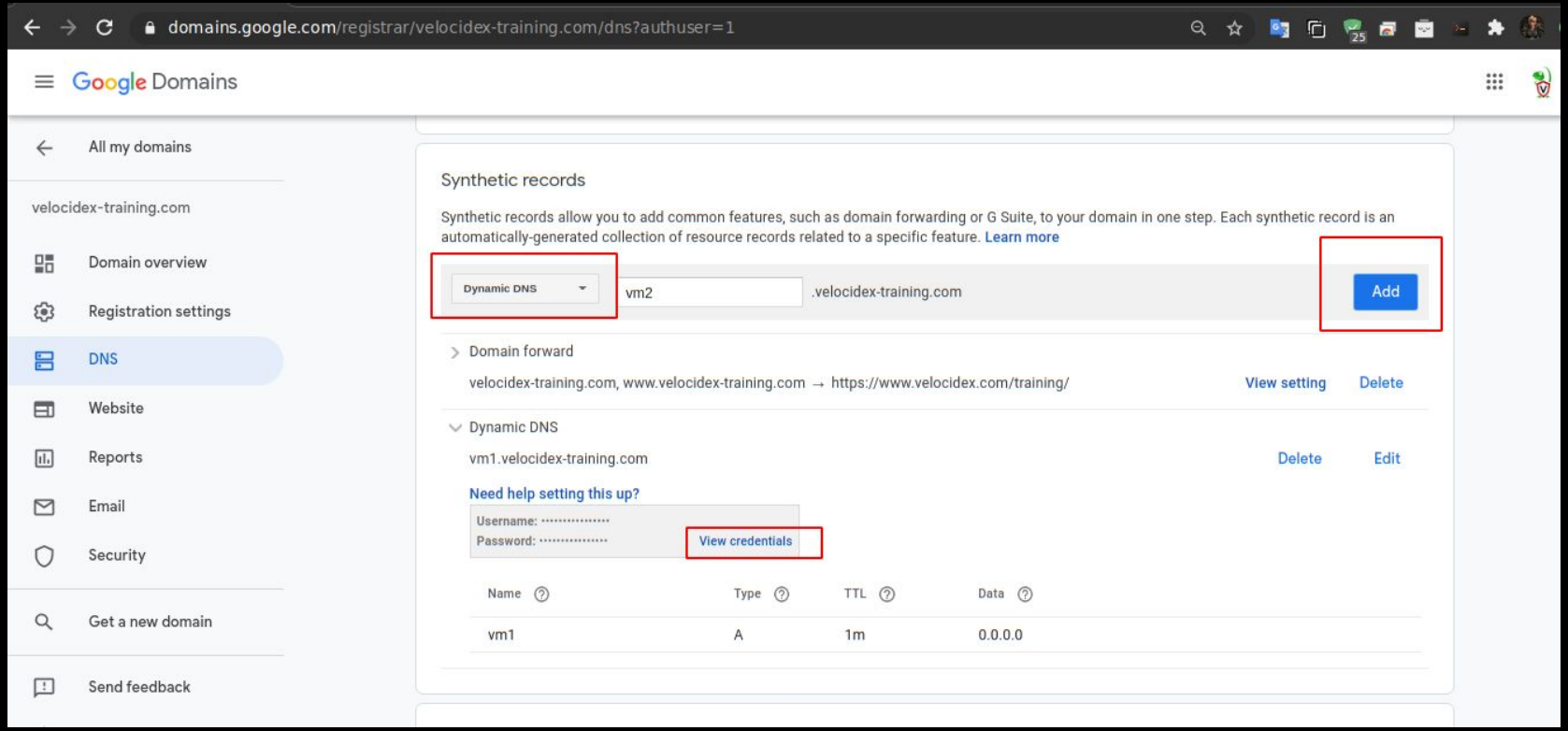
- Frontend served using TLS on port 8000 (connected to clients)
- GUI uses basic authentication with usernames/passwords.
- GUI Served over loopback port 8889 (127.0.0.1)
  - By default not exposed to the network
  - You can use SSH tunneling to forward the GUI

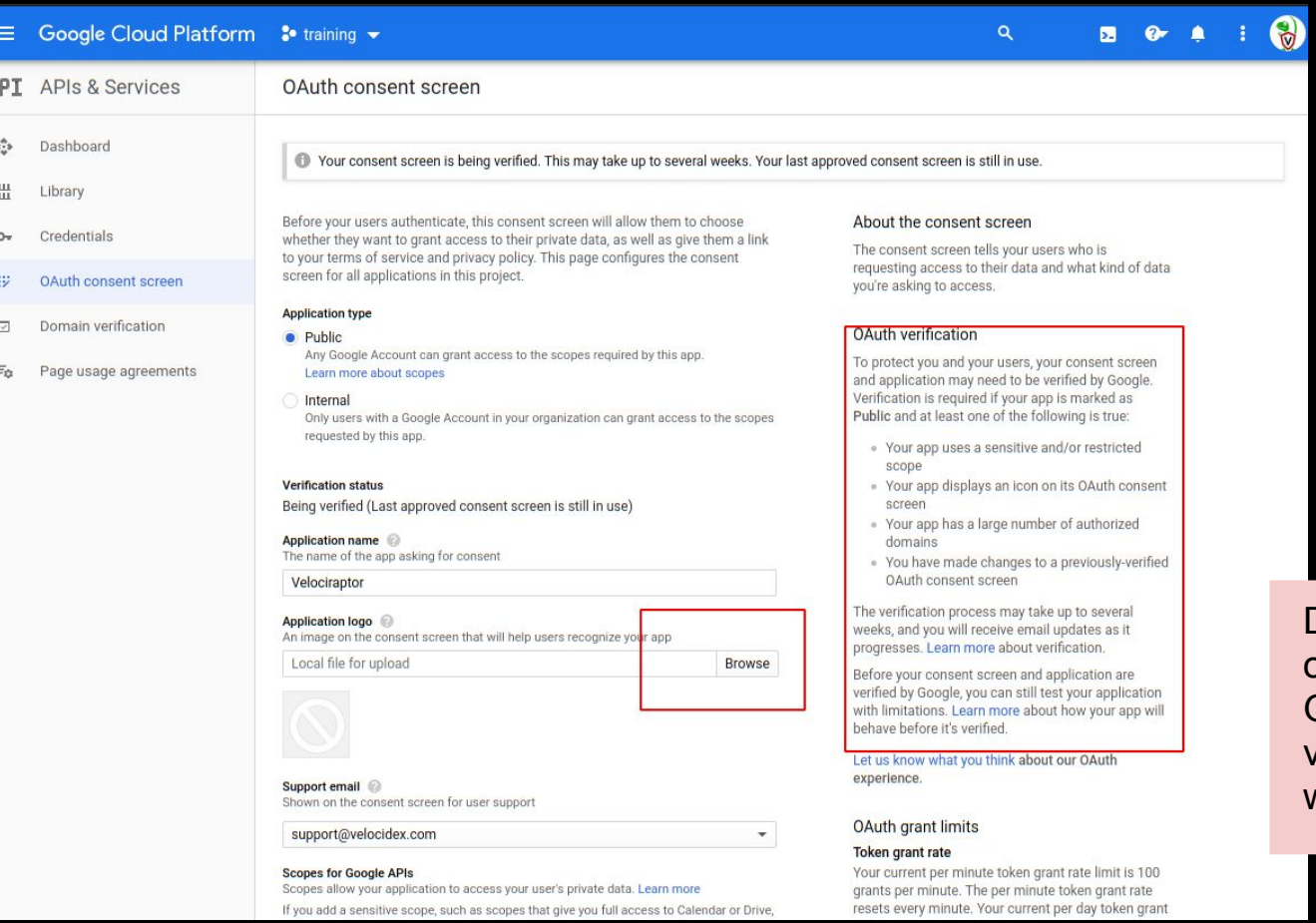
Steps to deploy Velociraptor

1. Provision a VM in the cloud
   a. Configure DNS (static or dynamic)
   b. Configure OAuth2 SSO
2. Generate configuration files
3. Build debian packages and install
4. Build MSI packages for Windows
5. Deploy via GPO/SCCM etc.

The instructor will demonstrate step 1. See the workshop setup document for credentials.

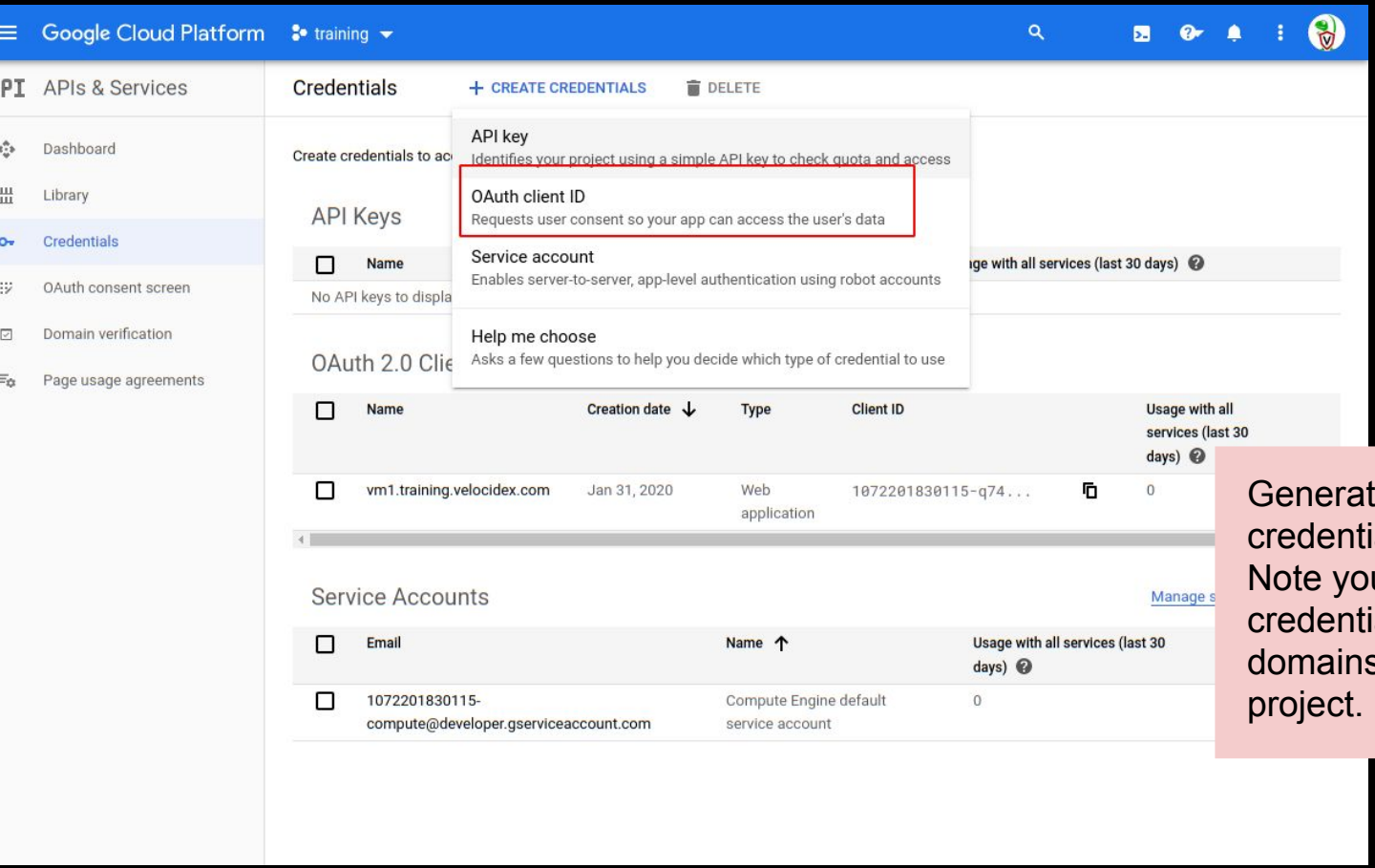
Setting Dynamic DNS with Google Domains



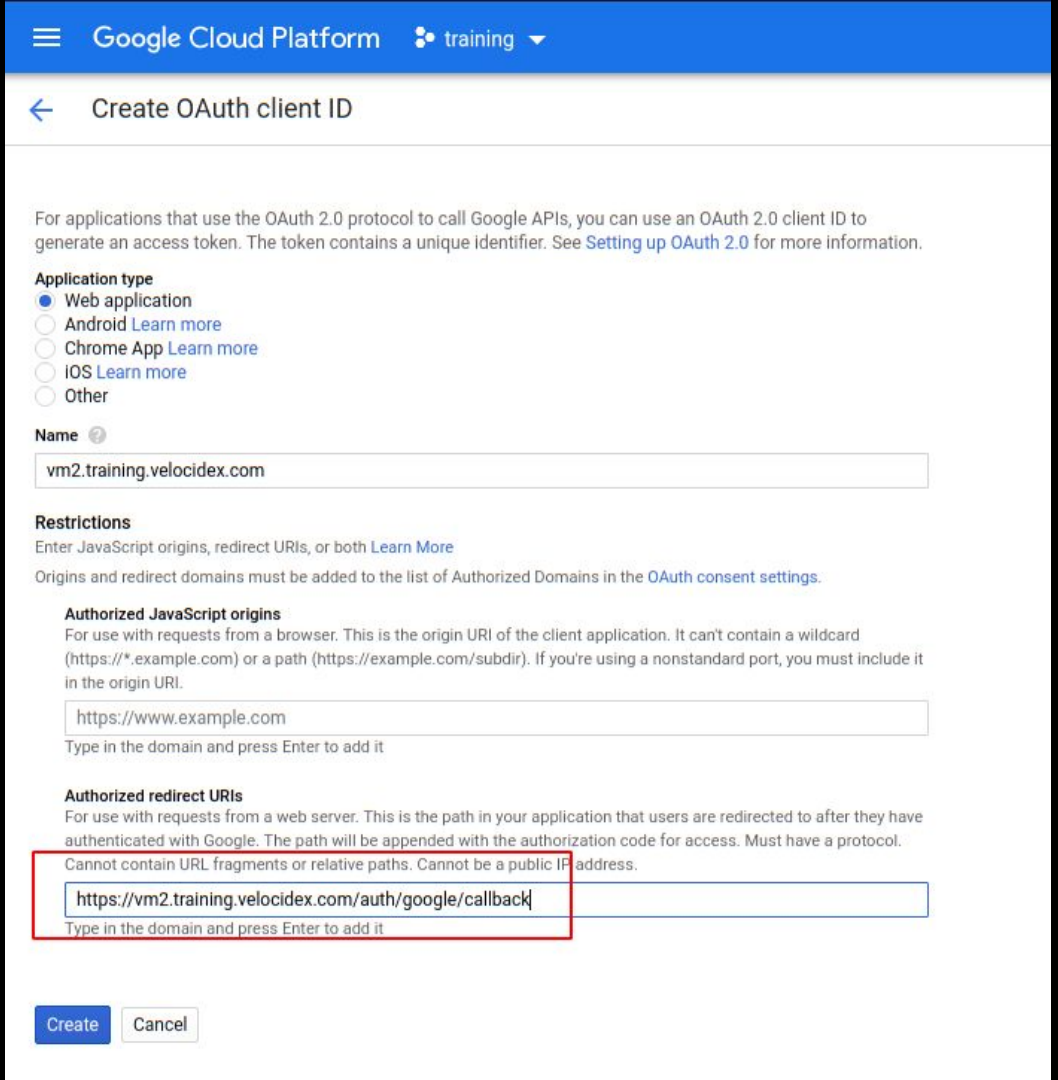


Configuring Google OAuth2 requires a new project and a consent screen

Do not add an application logo or require more permissions - Google will require OAuth verification which can take weeks!



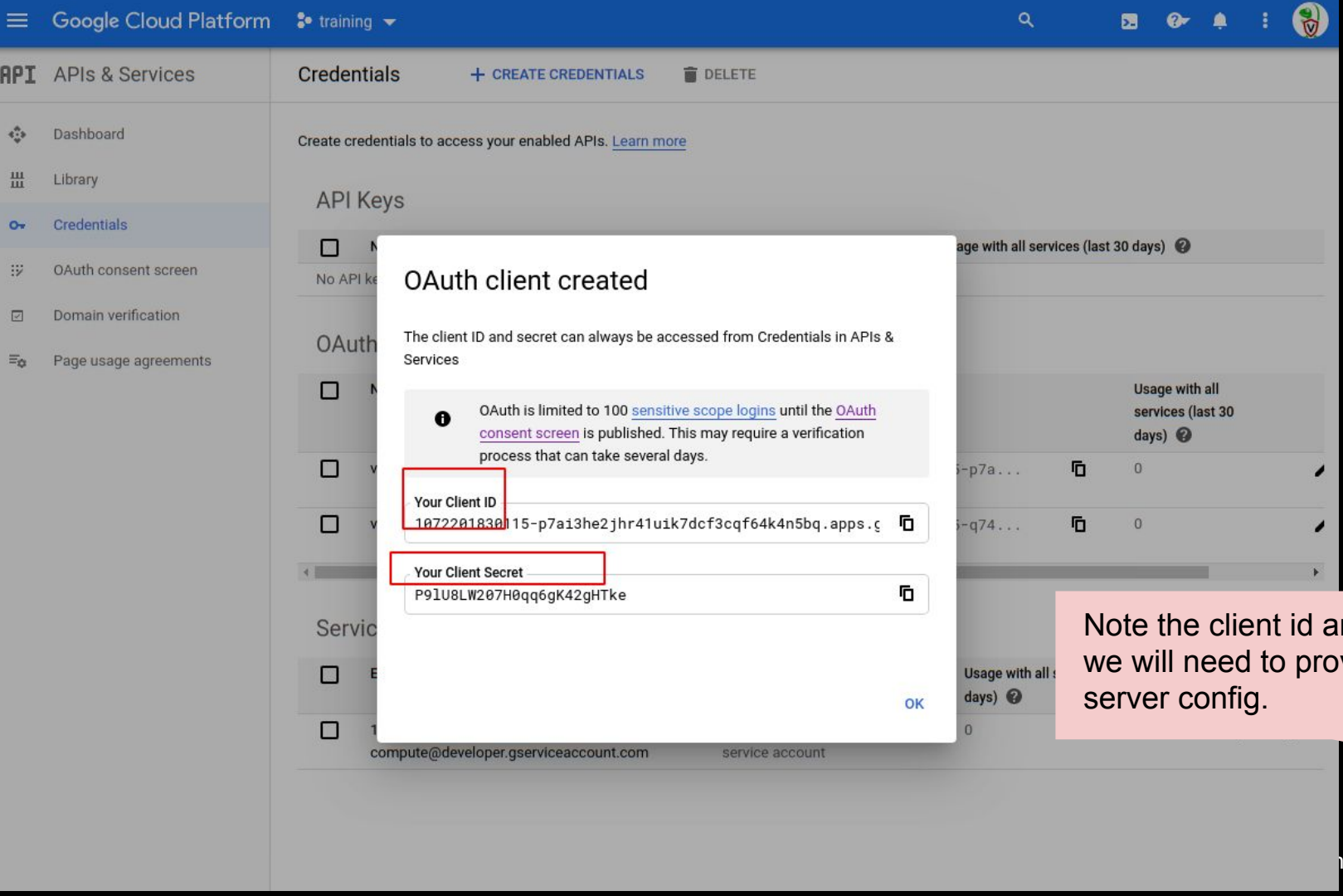
Generate OAuth client credentials.
Note you can have multiple credentials and multiple domains in the same GCP project.



The redirect URL is the url which Google will use to call back to Velociraptor with the user's successful login.

It must be

https://<domain>/auth/google/callback



Note the client id and secret - we will need to provide it in the server config.

nterprises

Installing a new server

Use the password provided in the Workshop setup to log into the server.

1. Fetch the latest Velociraptor Windows and Linux release binaries
2. Create a new configuration
   a. velociraptor config generate -i
3. Create a new server debian package
   a. velociraptor --config server.config.yaml debian server

Installing a new server

1. Push the debian package to the server using scp
   a. scp velociraptor_server*.deb mike@123.45.67.89:/tmp/

2. Install package
   a. sudo dpkg -i velociraptor_server*.deb

Latest release

v0.5.5-1

98119e5

Verified

Compare

Release 0.5.5-1

scudette released this 9 days ago · 9 commits to master since this release

This is a bugfix release from 0.5.5. Thanks for the bug reports and feedback.

Major issues fixed:

1. Memory leak in foreach() plugin
2. Python gRPC API handler crash
3. GUI Fix welcome screen logo was shown with incorrect size
4. GUI Fix VFS browser showing paths with % in their name
5. File based merge sort would fix memory issue on large ORDER BY queries.

Assets 7

- velociraptor-v0.5.5-1-darwin-amd64
- velociraptor-v0.5.5-1-linux-amd64
- velociraptor-v0.5.5-1-windows-386.exe
- velociraptor-v0.5.5-1-windows-amd64.exe
- velociraptor-v0.5.5-1-windows-amd64.msi
- Source code (zip)
- Source code (tar.gz)

Velocidex Enterprises

```
C:\Users\mike\Downloads>velociraptor-v0.4.3.3-windows-amd64.exe config generate -i
? Please select the datastore implementation
  FileBaseDataStore
? Path to the datastore directory. /data/
?
Welcome to the Velociraptor configuration generator
---------------------------------------------------

I will be creating a new deployment configuration for you. I will
begin by identifying what type of deployment you need.

  Authenticate users with Google OAuth SSO
? What is the public DNS name of the Frontend (e.g. www.example.com): [? for help] (www.example.com) vm1.training.velociraptor.velocidex.com
? What is the public DNS name of the Frontend (e.g. www.example.com): vm1.training.velocidex.com
? Enter the Google OAuth Client ID? 1072201830115-q74o1slmu2u3[redacted].googleusercontent.com
? Enter the Google OAuth Client Secret? 4VHuIw[redacted]SWTUg
? Are you using Google Domains DynDNS? Yes
? Google Domains DynDNS Username iYayz1Lj[redacted]tT
? Google Domains DynDNS Password 2Xst[redacted]
? GUI Username or email address to authorize (empty to end):
? Path to the logs directory. /data/logs
? Where should i write the server config file? server.config.yaml
? Where should i write the client config file? client.config.yaml
```

Generate new configuration with the details in the Workshop setup document.

Make sure to use /data/ as this will run on Linux

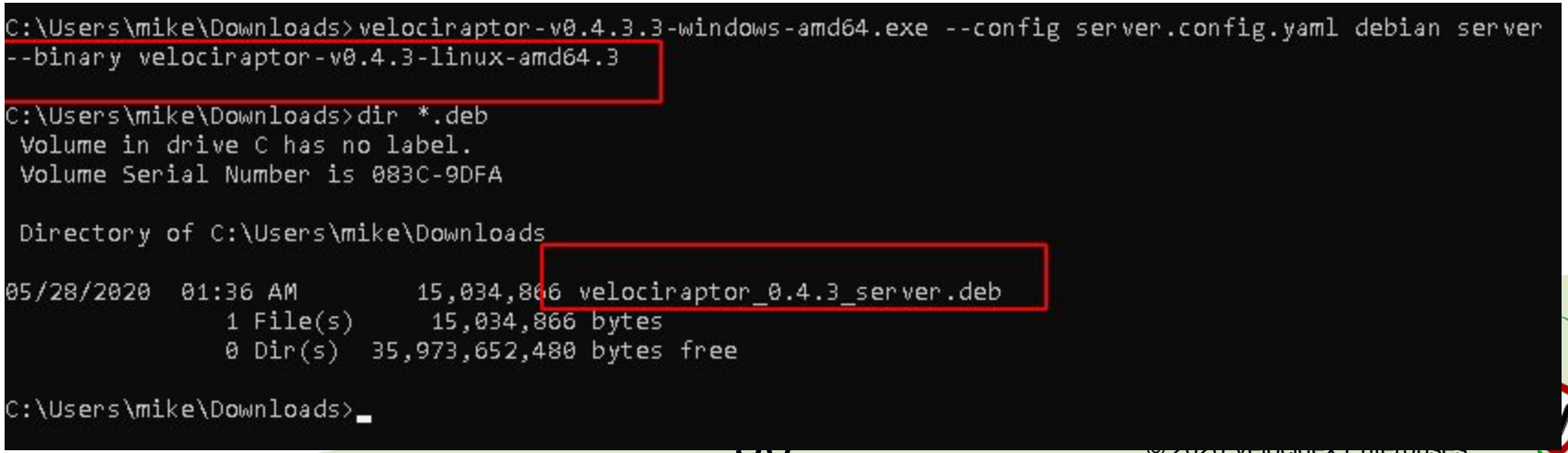
Automating config generation

- Some people want to automate the config generation step.
- Velociraptor supports a JSON merge for non interactive configuration generation

```
velociraptor config generate --merge
    '{"autocert_domain": "domain.com", "autocert_cert_cache": "/foo/bar"}'
```


Building a server deb package

```
./velociraptor-v0.5.5-windows.exe --config
~/server.config.yaml debian server
--binary velociraptor-v0.5.5-windows.exe
```



Deploying the server

scp the deb file to the target server

```
C:\Users\mike\Downloads>scp velociraptor_0.4.3_server.deb mike@34.71.243.48:
The authenticity of host '34.71.243.48 (34.71.243.48)' can't be established.
ECDSA key fingerprint is SHA256:OaQExNPmx95YD9qZFl6YGZT7ML5dFbj1XwMtgPsuY9Q.
Are you sure you want to continue connecting (yes/no)?
Warning: Permanently added '34.71.243.48' (ECDSA) to the list of known hosts.
mike@34.71.243.48's password:
velociraptor_0.4.3_server.deb                                 100%   14MB  70.6MB/s   00:00
```


```
Last login: Thu May 28 01:13:39 2020 from 74.125.41.33
mike@vm1-training:~$ sudo dpkg -i ./velociraptor_0.4.3_server.deb
Selecting previously unselected package velociraptor-server.
(Reading database ... 37691 files and directories currently installed.)
Preparing to unpack .../velociraptor_0.4.3_server.deb ...
Unpacking velociraptor-server (0.4.3) ...
dpkg: dependency problems prevent configuration of velociraptor-server:
 velociraptor-server depends on libcap2-bin; however:
  Package libcap2-bin is not installed.

dpkg: error processing package velociraptor-server (--install):
 dependency problems - leaving unconfigured
Errors were encountered while processing:
 velociraptor-server
mike@vm1-training:~$ sudo apt-get install -f
Reading package lists... Done
Building dependency tree
Reading state information... Done
Correcting dependencies... Done
The following additional packages will be installed:
  libcap2-bin libpam-cap
The following NEW packages will be installed:
  libcap2-bin libpam-cap
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
1 not fully installed or removed.
Need to get 40.0 kB of archives.
After this operation, 128 kB of additional disk space will be used.
Do you want to continue? [Y/n] _
```

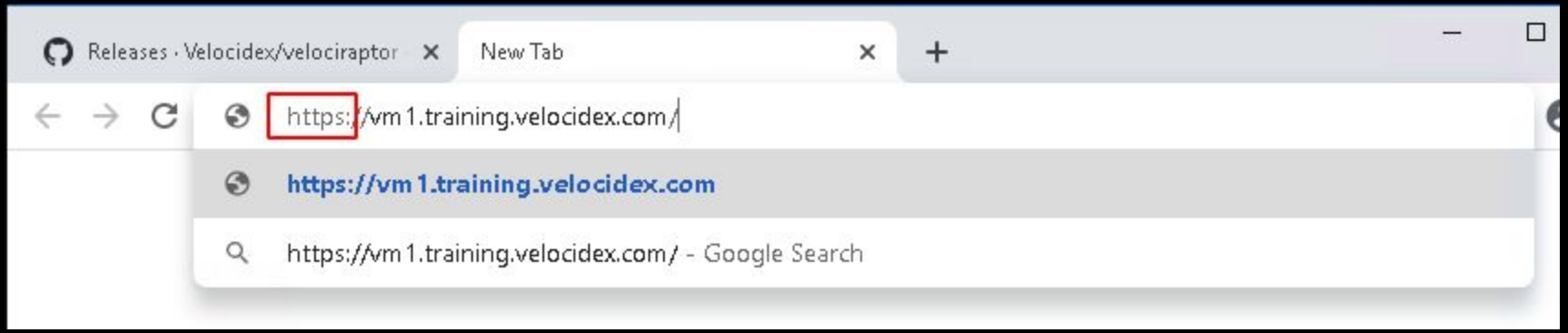
When installing the deb package you might need to install dependencies by using "apt-get install -f"

```
Setting up libcap2-bin (1:2.25-1) ...
Setting up velociraptor-server (0.4.3) ...
Adding group `velociraptor' (GID 112) ...
Done.
Adding system user `velociraptor' (UID 108) ...
Adding new user `velociraptor' (UID 108) with group `velociraptor' ...
Not creating home directory `/etc/velociraptor/'.
Created symlink /etc/systemd/system/multi-user.target.wants/velociraptor_server.service → /etc/systemd/sys
tem/velociraptor_server.service.
Processing triggers for man-db (2.7.6.1-2) ...
mike@vm1-training:~$ sudo service velociraptor_server status
● velociraptor_server.service - Velociraptor linux amd64
   Loaded: loaded (/etc/systemd/system/velociraptor_server.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2020-05-28 01:43:35 UTC; 36s ago
 Main PID: 1556 (velociraptor)
    Tasks: 9 (limit: 4915)
   CGroup: /system.slice/velociraptor_server.service
           ├─1556 /bin/bash /usr/local/bin/velociraptor --config /etc/velociraptor/server.config.yaml fron
tend
           └─1557 /usr/local/bin/velociraptor.bin --config /etc/velociraptor/server.config.yaml frontend

May 28 01:43:35 vm1-training systemd[1]: Started Velociraptor linux amd64.
mike@vm1-training:~$
```

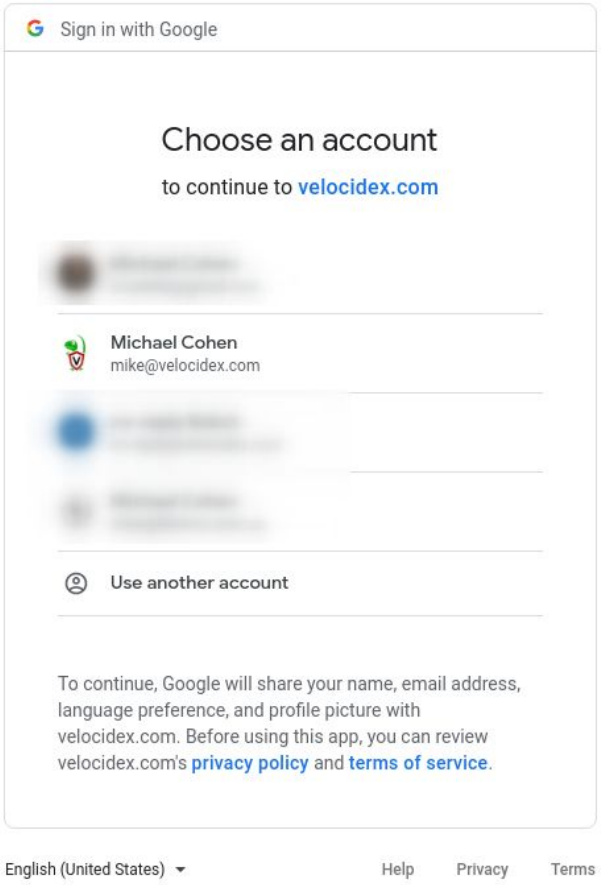
The service adds a new velociraptor user to run under. You can now access the Velociraptor server using your browser.

```
ic@velotest:~$ ./velociraptor-v0.4.3-linux-amd64 --config server.config.yaml debian server
ic@velotest:~$ sudo dpkg -i velociraptor_0.4.2_server.deb
(Reading database ... 39245 files and directories currently installed.)
Preparing to unpack velociraptor_0.4.2_server.deb ...
Removed /etc/systemd/system/multi-user.target.wants/velociraptor_server.service.
Unpacking velociraptor-server (0.4.2) over (0.4.1) ...
Setting up velociraptor-server (0.4.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/velociraptor_server.service → /etc/systemd/system/velociraptor_server.service

ic@velotest:~$ sudo service velociraptor_server status
● velociraptor_server.service - Velociraptor linux amd64
   Loaded: loaded (/etc/systemd/system/velociraptor_server.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-05-11 13:36:33 UTC; 15s ago
 Main PID: 5492 (velociraptor)
    Tasks: 9 (limit: 4915)
   CGroup: /system.slice/velociraptor_server.service
           ├─5492 /bin/bash /usr/local/bin/velociraptor --config /etc/velociraptor/server.config.yaml frontend
           └─5493 /usr/local/bin/velociraptor.bin --config /etc/velociraptor/server.config.yaml frontend

May 11 13:36:33 velotest systemd[1]: Started Velociraptor linux amd64.
```

Build a Debian package using the new configuration file.
Install the package
Check the new service is running properly.



The first time you navigate to the SSL URL the server will obtain a certificate from Let's Encrypt. There will be a small pause as this happens.



You will be redirected to Google for authentication - Velociraptor does not handle any credentials in this configuration. Google will determine if the user authenticated properly (2 FA etc) and convey simple info like the user's email address and avatar.

User permissions

When running the deb package Velociraptor is running as a non-root user with limited permissions. You must change to this user before manipulating any data, or the service may not be able to open the modified files.

- Velociraptor will refuse running as another user or as root to prevent permission problems

sudo -u velociraptor ...

```
mic@velotest:~$ velociraptor user add joe@example.com --role reader
'/etc/velociraptor/server.config.yaml' is not readable, you will need to run this as the velociraptor user ('sudo -u velociraptor bash').
mic@velotest:~$ sudo velociraptor user add joe@example.com --role reader
velociraptor.bin: error: Velociraptor should be running as the 'velociraptor' user but you are 'root'. Please change user with sudo first
.
mic@velotest:~$ sudo -u velociraptor velociraptor user add joe@example.com --role reader
Authentication will occur via Google - therefore no password needs to be set.
mic@velotest:~$
```

Role based Access Control

Velociraptor uses a simple role based access control scheme for now

- Various Actions require specific permissions
- Users are granted roles which bestow them with a set of permissions.

Granting a user role

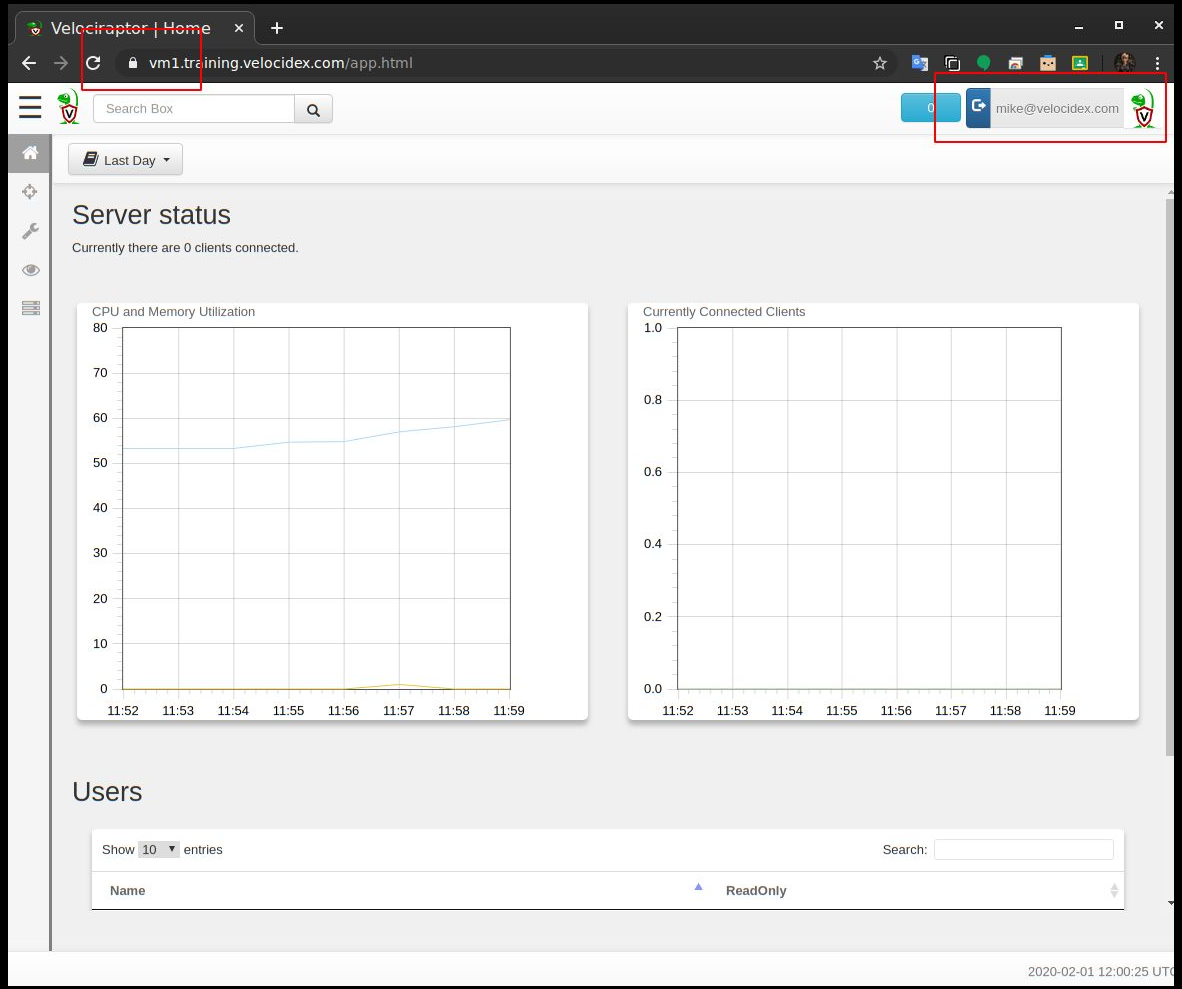
1. Currently roles are hard coded
   a. **administrator** - Can do anything without limits
   b. **reader** - Can read collected data and notebooks
   c. **api** - Can connect over the API (more later)
   d. **analyst** - reader + create bulk downloads, edit notebooks
   e. **investigator** - analyst + schedule new collections and hunts
   f. **artifact_writer** - powerful role that allows the user to create and modify artifacts (more on this later)

Just because a user is authenticated by Google does not mean they have access to the Velociraptor console!

You must authorize each user to access the console by granting them at least the **reader** role.

Manipulate acls using the "acl show" "acl grant" command

```
elociraptor@velotest:/home/mic$ velociraptor acl grant mike@velocidex.com --role reader,investigator
elociraptor@velotest:/home/mic$ velociraptor acl show mike@velocidex.com
"roles":["reader","investigator"]}
elociraptor@velotest:/home/mic$ velociraptor acl show --effective mike@velocidex.com
"any_query":true,"read_results":true,"label_clients":true,"collect_client":true,"notebook_editor":true,"prepare_results":true}
elociraptor@velotest:/home/mic$
```



Your Velociraptor server is ready.

You should have a valid SSL Cert and Avatar provided by Google OAuth2

Velociraptor internals

Digging into Velociraptor!

The file store

Velociraptor uses a filestore abstraction to store data. By default, we use a simple directory structure in the filesystem.

- Having simple files simplifies data retention, data migration, backups etc.
- Makes it easy to integrate with another system (use scp or rsync to just copy files around).
- If files are deleted, Velociraptor will just recreate them - it is safe to just remove everything!

Now let’s configure some clients.

Deploying clients

We typically distribute signed MSI packages which include the client's config file inside them.
This makes it easier to deploy as there is only one package to install.

We also change name of service/binary etc to make the service a little bit harder to stop.

Deploying clients

It is possible to embed the config in the clients using the **velociraptor config repack** command (more later)

Pros

- Only a single binary no need for an additional config file

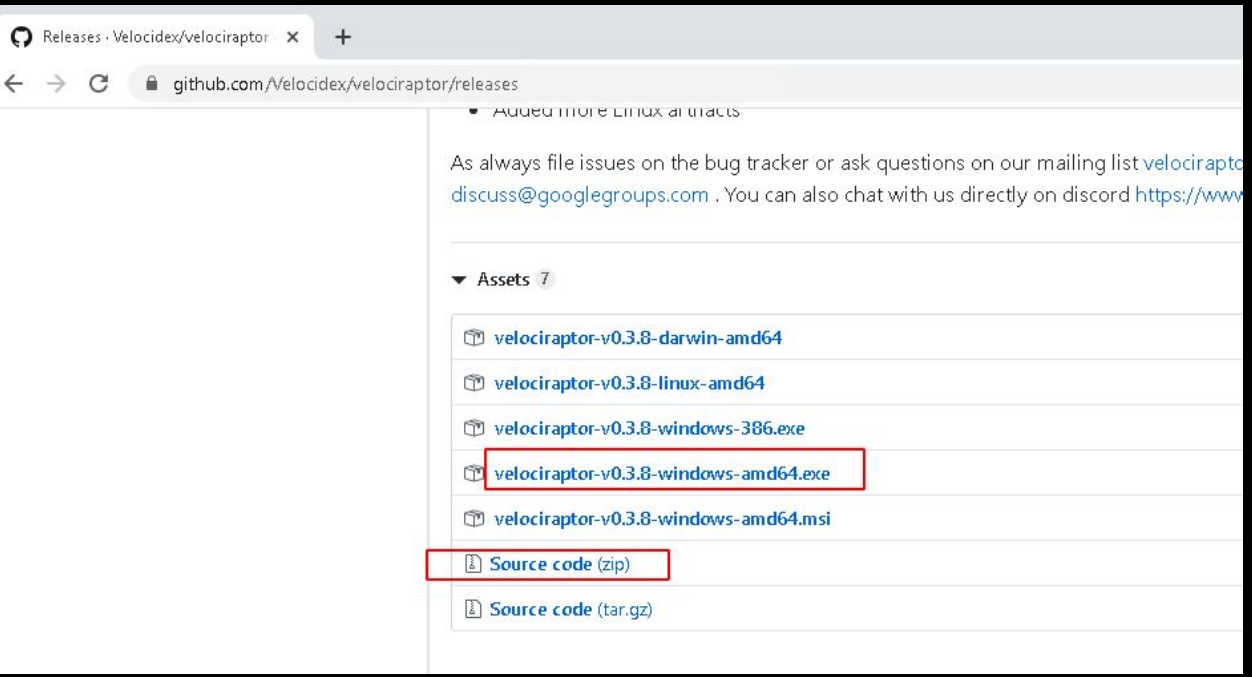
Cons

- You have to sign the binary again since the config alters the binary.

Resigning binaries

After buying a code signing cert you can use a script to sign automatically.
We recommend having a standalone isolated signing machine or VM with FDE

```
#!/bin/bash

osslsigncode sign -pkcs12 ~/private/code_sign.pkcs12 -n "Velociraptor" \
             -h sha2  -t http://timestamp.verisign.com/scripts/timstamp.dll \
             -i https://www.velocidex.com/ \
             -in "$1" -out "$1.signed.exe" -askpass

mv "$1.signed.exe" "$1"
```



On your windows machine, Download the latest binary and the source code.

github.com/velocidex/velociraptor/releases

Velociraptor's public directory

It is handy to have somewhere to serve files from. Velociraptor has a public directory where files are served **without any authentication requirements**

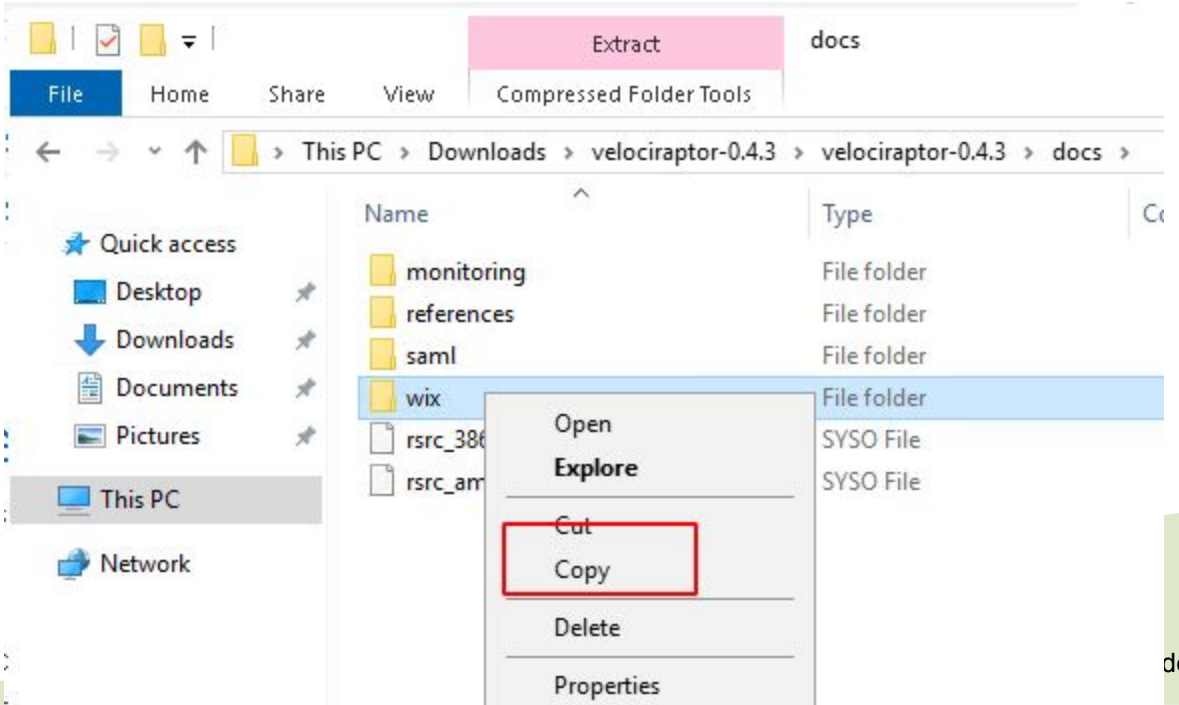
- ❏ We can use this to distribute third party binaries
- ❏ We can serve velociraptor MSI files
- ❏ We can serve various support files (yara rules etc).

Velociraptor's public directory

Select the Admin.Client.Upgrade artifact and upload the MSI to the tools setup page (We will learn about that in the next few sessions).

This will now produce a random URL you can serve the MSI from.

Copy WIX source to desktop.



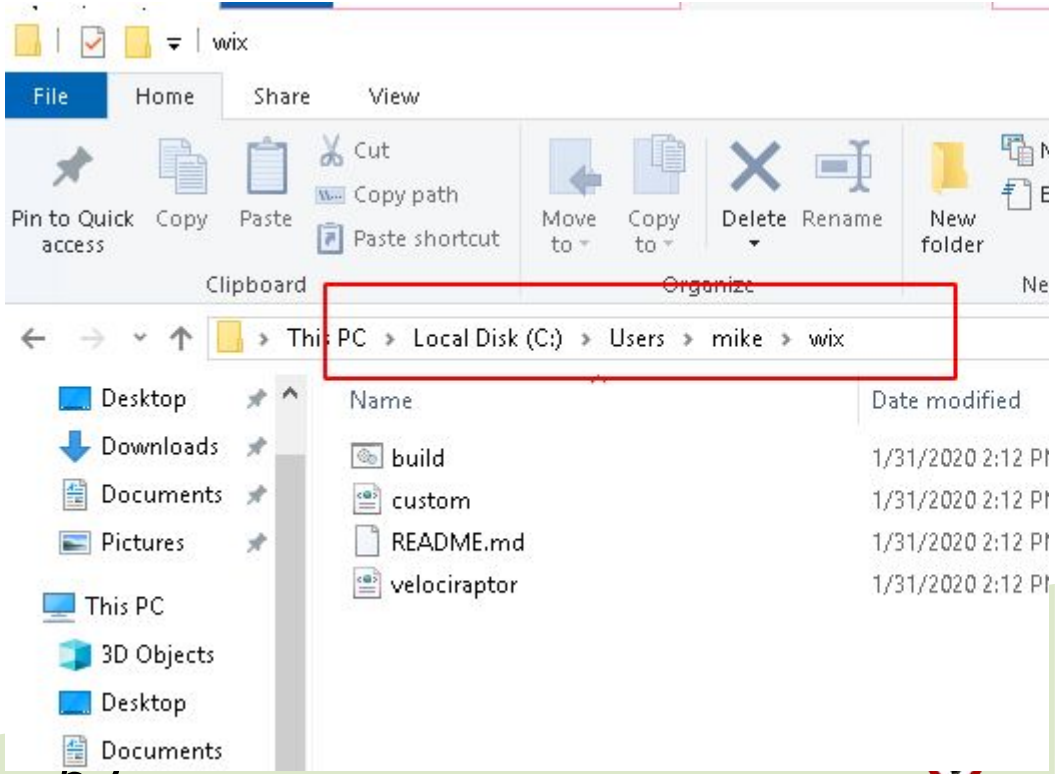
dex Enterprises

Build an MSI using Wix Toolkit

Extract the docs/wix directory from the Velociraptor source tree.

These are the required files to construct a new MSI

The main file we use is custom.xml . This file will embed the config file within the MSI and deploy it to the correct directory.



:\wix\release\velociraptor.xml - Notepad++

Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

elociraptor.xml

```
<?xml version='1.0' encoding='windows-1252'?>
<?define AppRegKey="Software\Velocidex\Velociraptor" ?>
<?define PackageDescription="Velociraptor Service Installer" ?>
<?define Manufacturer="Velocidex" ?>
<?define Name="Velociraptor" ?>
<?define Version="0.50.1" ?>
<?define BinaryName="Velociraptor.exe" ?>

<Wix xmlns="http://schemas.microsoft.com/wix/2006/wi"
     xmlns:util="http://schemas.microsoft.com/wix/UtilExtension"
     >
  <Product Name='$(var.Name)' Manufacturer='$(var.Manufacturer)'
           Id='*'
           UpgradeCode='82E586E1-1700-4041-9042-8946BE19B69F'
           Language='1033' Codepage='1252' Version='$(var.Version)'>
    <Package Id='*' Keywords='Installer' Description="$(var.PackageDescription)"
             Comments='$(var.PackageDescription)'
             Manufacturer='$(var.Manufacturer)'
             InstallerVersion='200' Languages='1033' Compressed='yes'
             SummaryCodepage='1252' />
    <Media Id='1' Cabinet='Sample.cab' EmbedCab='yes' DiskPrompt='CD-ROM #1' />
    <Property Id='DiskPrompt' Value="Installation [1]" />

    <Directory Id='TARGETDIR' Name='SourceDir'>
      <Directory Id='ProgramFiles64Folder' Name='PFiles'>
        <Directory Id='INSTALLDIR' Name='$(var.Name)'>
                  <Directory Id="CACHEDIR" Name="Tools">
                    <Component Id="Tools" Guid="97dc953a-8a2f-494f-9585-56ae526d0b48">
                          <CreateFolder />
                        </Component>
                  </Directory>
          <Component Id='MainExecutable'
```

There are many knobs to tweak here

- The name of the binary
- The location of the files
- The name of the service
- The name of the config file.

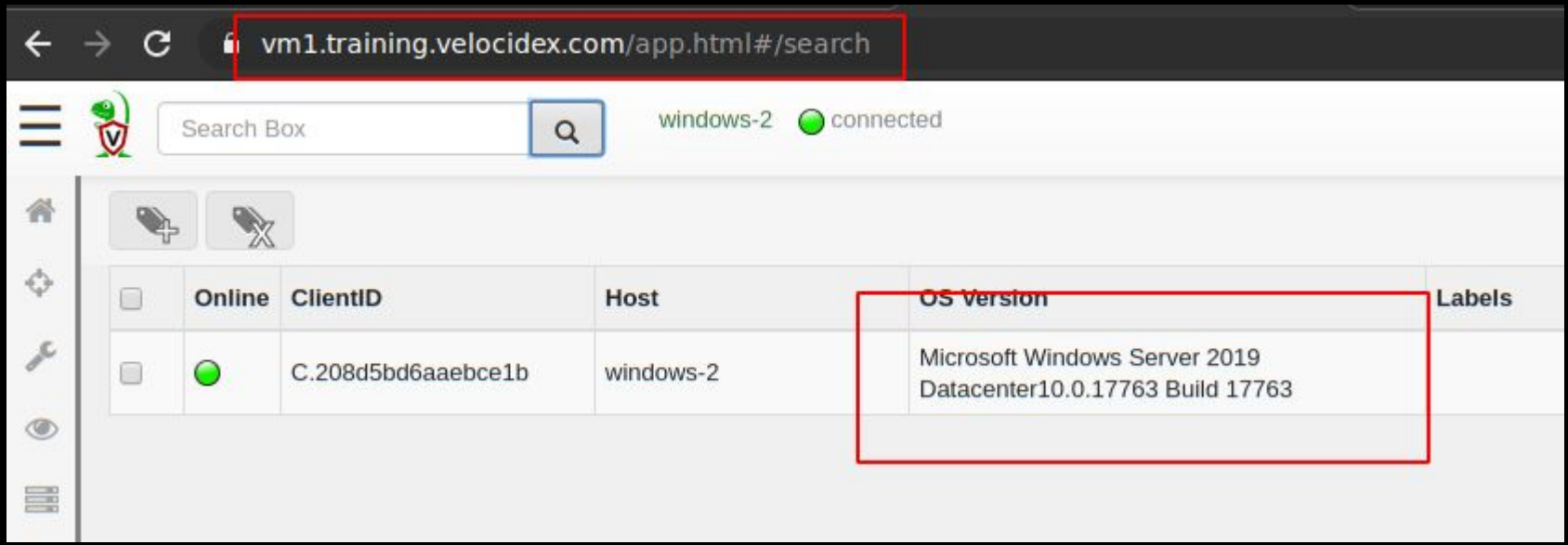
WIX will take the binary and config file from the Output directory, so create it and place the files there.

```
C:\Windows\system32>cd \Users\mike\Desktop\wix

C:\Users\mike\Desktop\wix>dir
 Volume in drive C has no label.
 Volume Serial Number is 083C-9DFA

 Directory of C:\Users\mike\Desktop\wix

05/28/2020  02:02 AM    <DIR>          .
05/28/2020  02:02 AM    <DIR>          ..
05/28/2020  02:02 AM               304 buid_x86.bat
05/28/2020  02:02 AM               314 build.bat
05/28/2020  02:02 AM               302 build_custom.bat
05/28/2020  02:02 AM               310 build_x86_custom.bat
05/28/2020  02:02 AM             3,004 custom.xml
05/28/2020  02:02 AM             3,006 custom_x86.xml
05/28/2020  02:02 AM             2,473 README.md
05/28/2020  02:02 AM             2,766 velociraptor.xml
               8 File(s)         12,479 bytes
               2 Dir(s)  35,690,766,336 bytes free

C:\Users\mike\Desktop\wix>mkdir output

C:\Users\mike\Desktop\wix>copy ..\..\Downloads\velociraptor-v0.4.3.3-windows-amd64.exe output\velociraptor.exe
        1 file(s) copied.

C:\Users\mike\Desktop\wix>copy ..\..\Downloads\client.config.yaml output\client.config.yaml
        1 file(s) copied.
```

```
C:\Users\mike\Desktop\wix>build_custom.bat

C:\Users\mike\Desktop\wix>"c:\Program Files (x86)\WiX Toolset v3.11\bin\candle.exe" custom.xml -arch x64 -ext "c:\Progr
m Files (x86)\WiX Toolset v3.11\bin\WixUtilExtension.dll"
Windows Installer XML Toolset Compiler version 3.11.2.4516
Copyright (c) .NET Foundation and contributors. All rights reserved.

custom.xml

C:\Users\mike\Desktop\wix>"c:\Program Files (x86)\WiX Toolset v3.11\bin\light.exe" custom.wixobj -ext "c:\Program Files
(x86)\WiX Toolset v3.11\bin\WixUtilExtension.dll"
Windows Installer XML Toolset Linker version 3.11.2.4516
Copyright (c) .NET Foundation and contributors. All rights reserved.


C:\Users\mike\Desktop\wix>dir *.msi
 Volume in drive C has no label.
 Volume Serial Number is 083C-9DFA

 Directory of C:\Users\mike\Desktop\wix

05/28/2020  02:06 AM        15,093,760 custom.msi
               1 File(s)     15,093,760 bytes
               0 Dir(s)  35,625,000,960 bytes free

C:\Users\mike\Desktop\wix>msiexec /i custom.msi
```

The custom msi contains the client config embedded in it.

This is the recommended way to deploy clients.



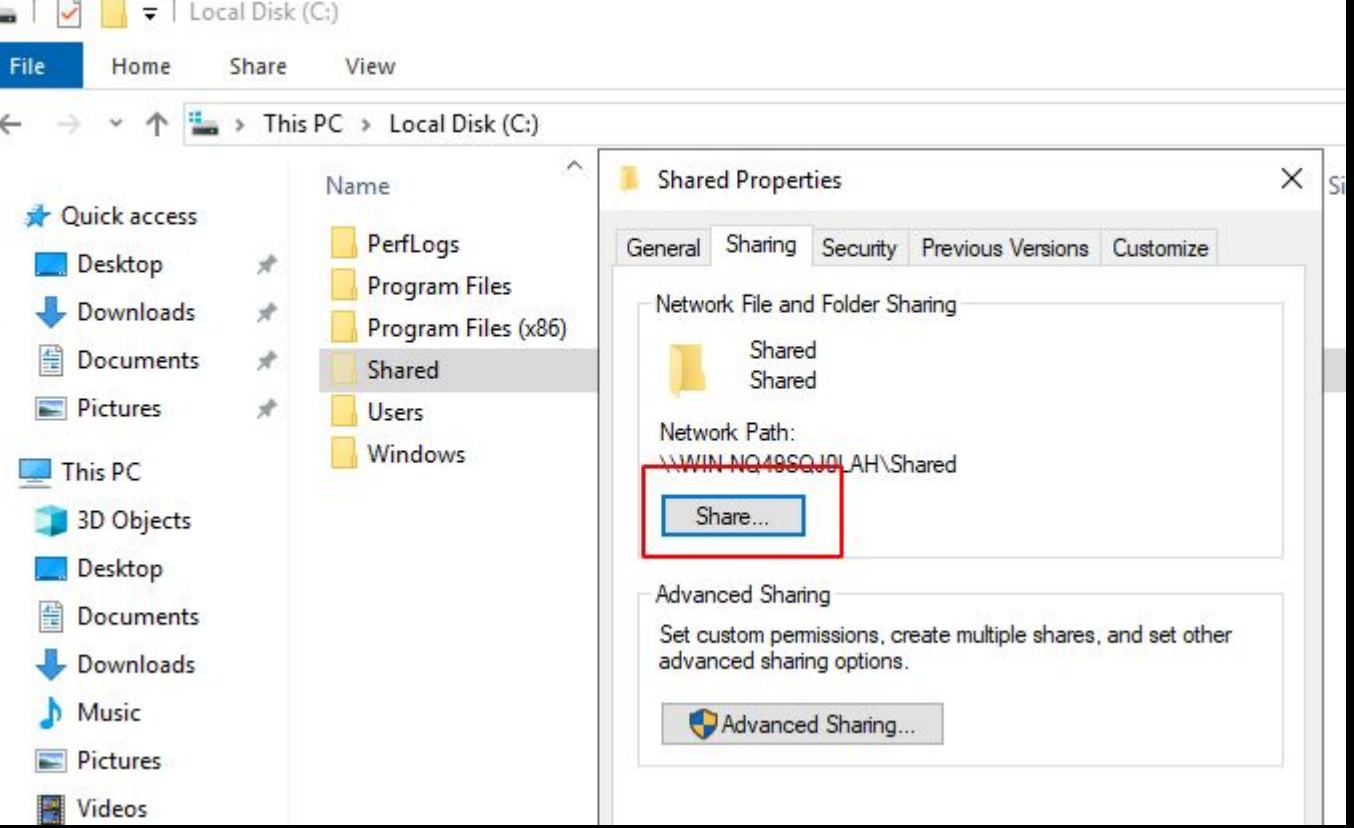
After installing the MSI you should be able to see it immediately in the server's search screen.

Domain deployment

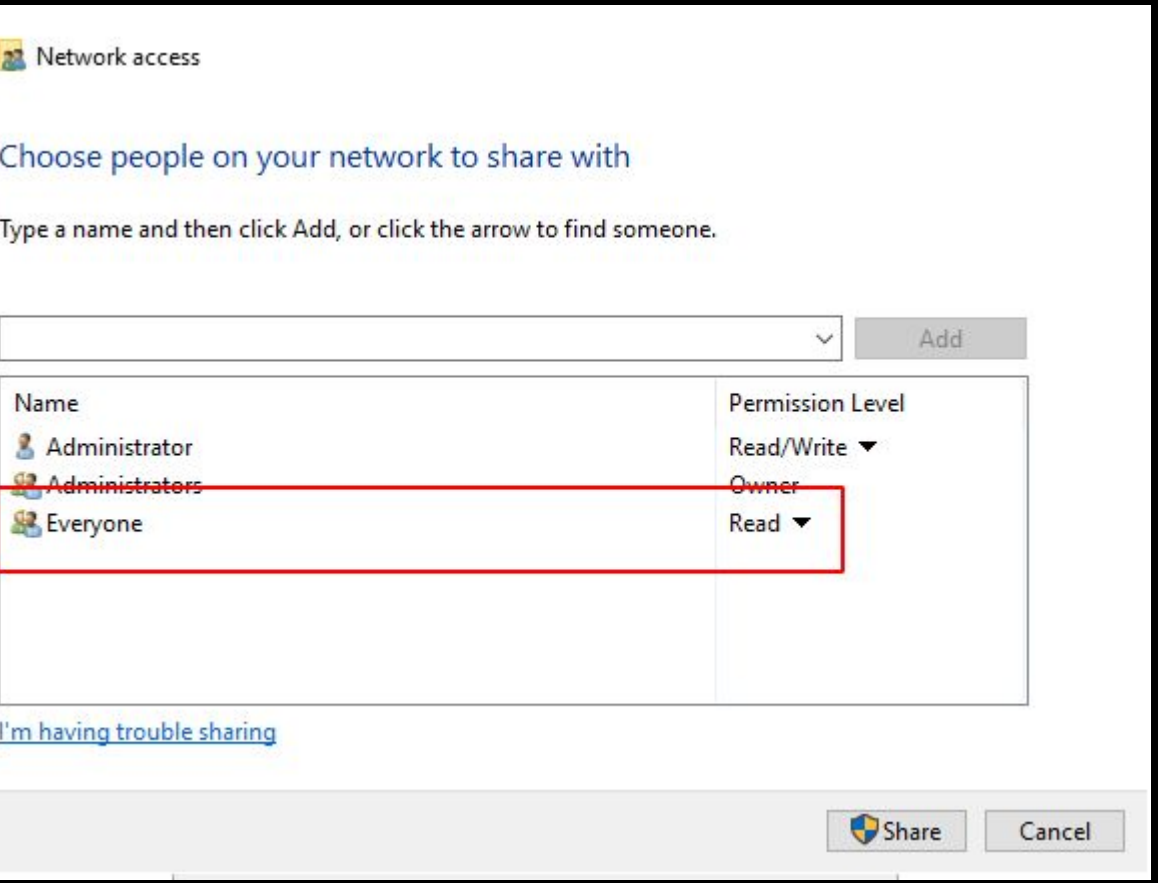
We can deploy the MSI to the entire domain using group policy.

2 Methods

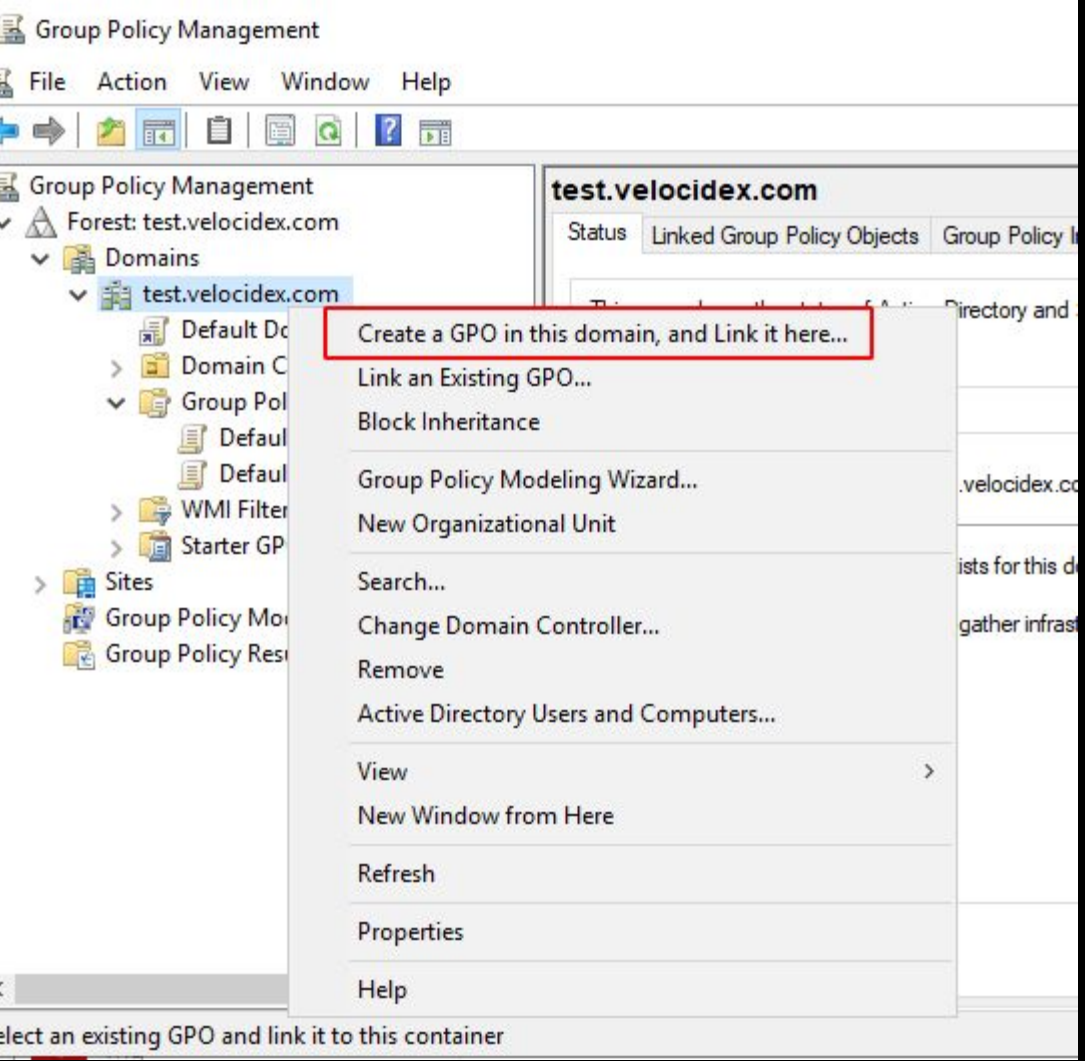
1. Via scheduled tasks.
2. Via assigned software.



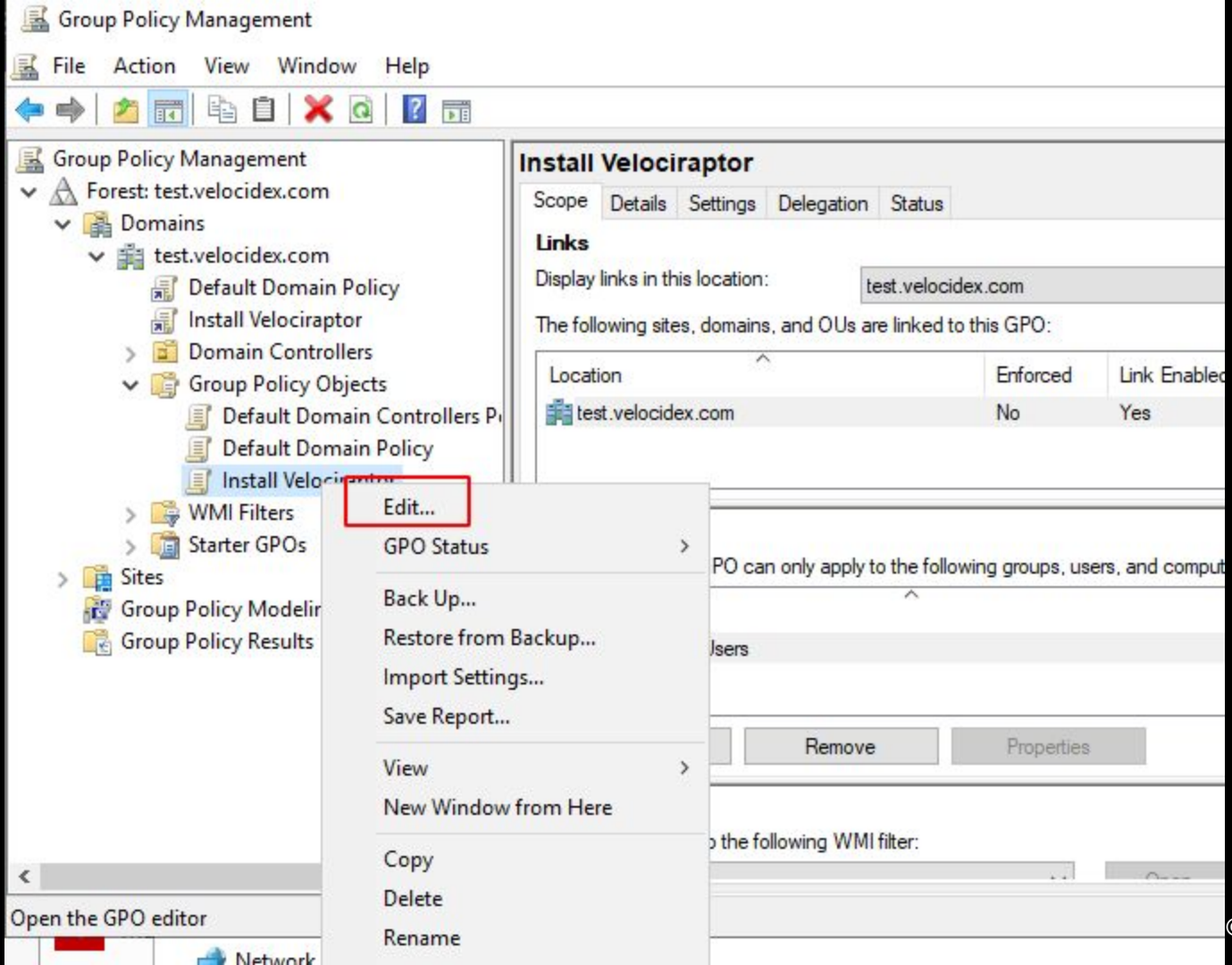
Create a share to serve the MSI from.



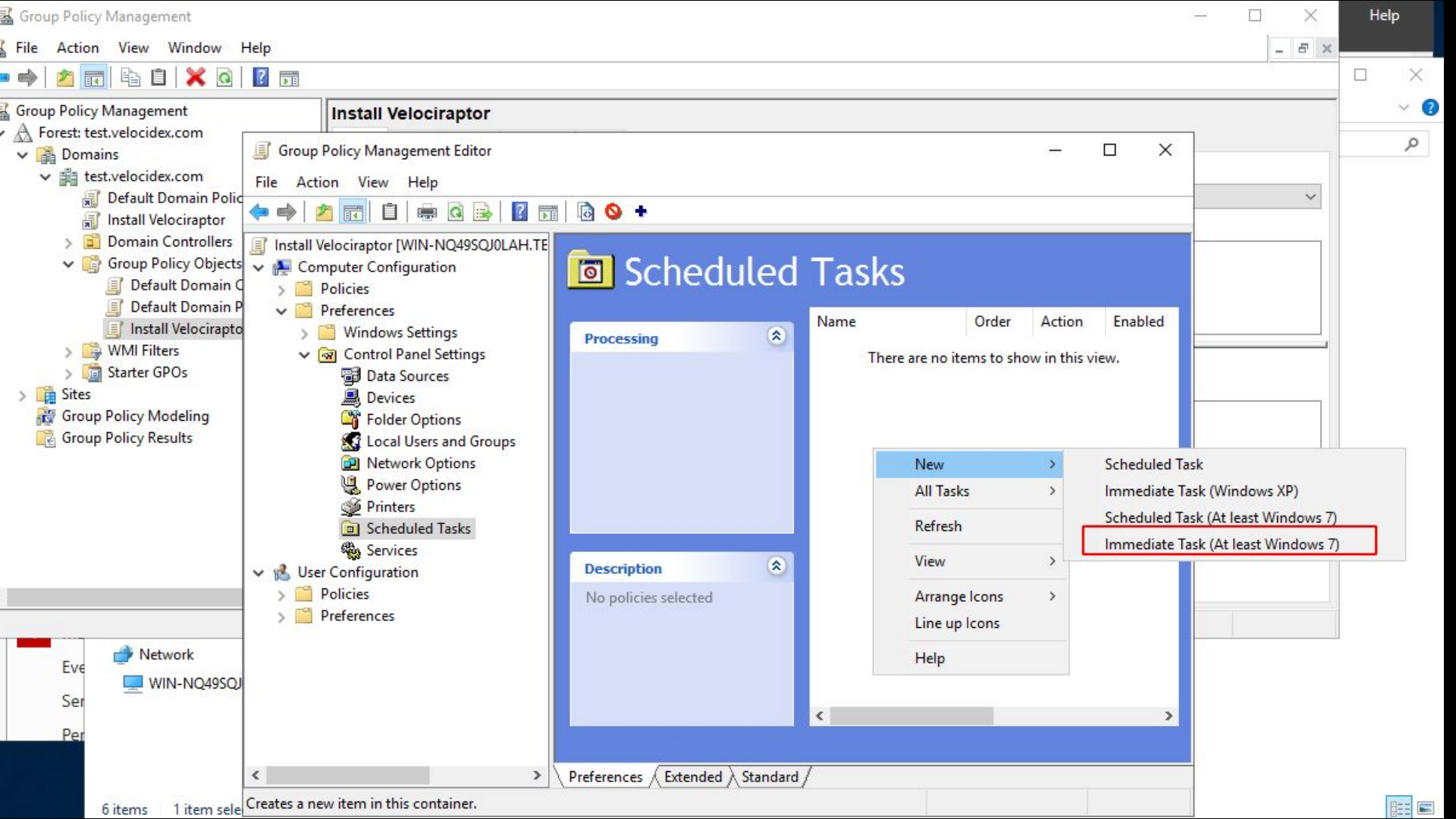
Ensure everyone has read access from this share - and only administrators have write access!

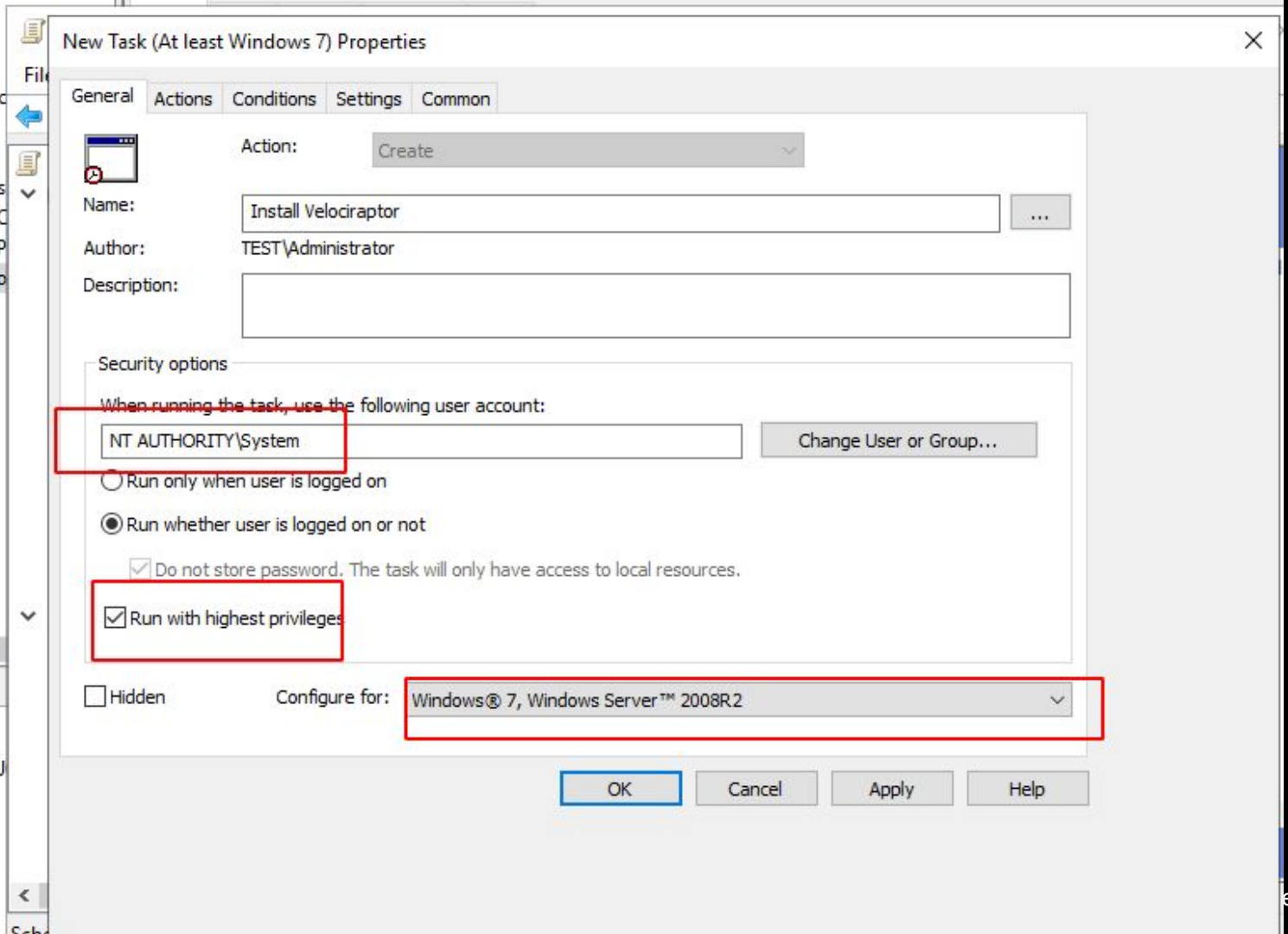


Use the group policy management tool create a new Group Policy Object in the domain (or OU)

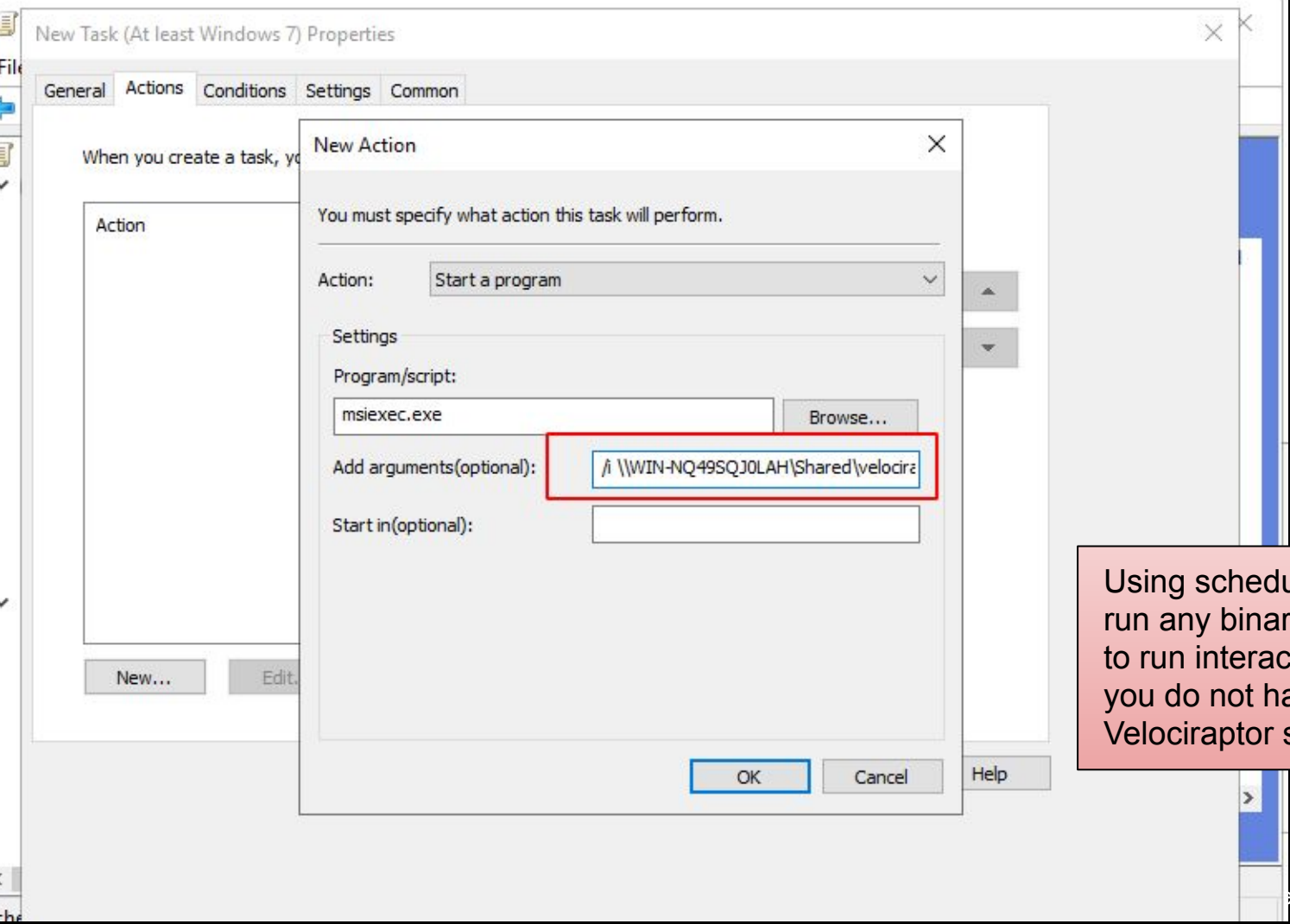


Edit the new GPO

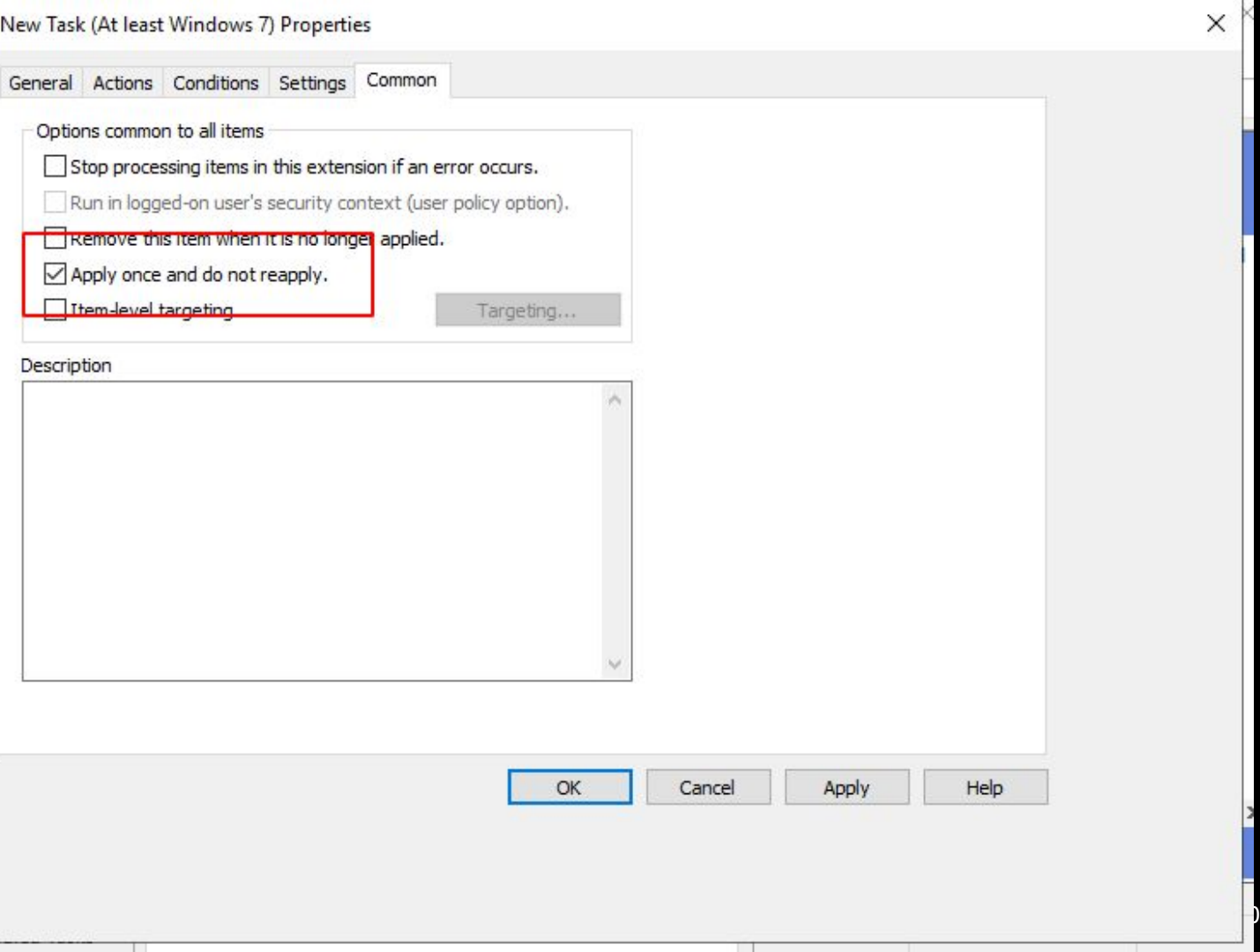

Group Policy Management
File Action View Window Help
Group Policy Management
Forest: test.velocidex.com
Domains
test.velocidex.com
Default Domain Polic
Install Velociraptor
Domain Controllers
Group Policy Objects
Default Domain C
Default Domain P
Install Velociraptor
WMI Filters
Starter GPOs
Sites
Group Policy Modeling
Group Policy Results
Install Velociraptor
Group Policy Management Editor
File Action View Help
Install Velociraptor [WIN-NQ49SQJ0LAH.TE
Computer Configuration
Policies
Preferences
Windows Settings
Control Panel Settings
Data Sources
Devices
Folder Options
Local Users and Groups
Network Options
Power Options
Printers
Scheduled Tasks
Services
User Configuration
Policies
Preferences
Scheduled Tasks
Processing
Description
No policies selected
Name
Order
Action
Enabled
There are no items to show in this view.
New
All Tasks
Refresh
View
Arrange Icons
Line up Icons
Help
Scheduled Task
Immediate Task (Windows XP)
Scheduled Task (At least Windows 7)
Immediate Task (At least Windows 7)
Preferences
Extended
Standard
Creates a new item in this container.
Network
WIN-NQ49SQJ
6 items
Help



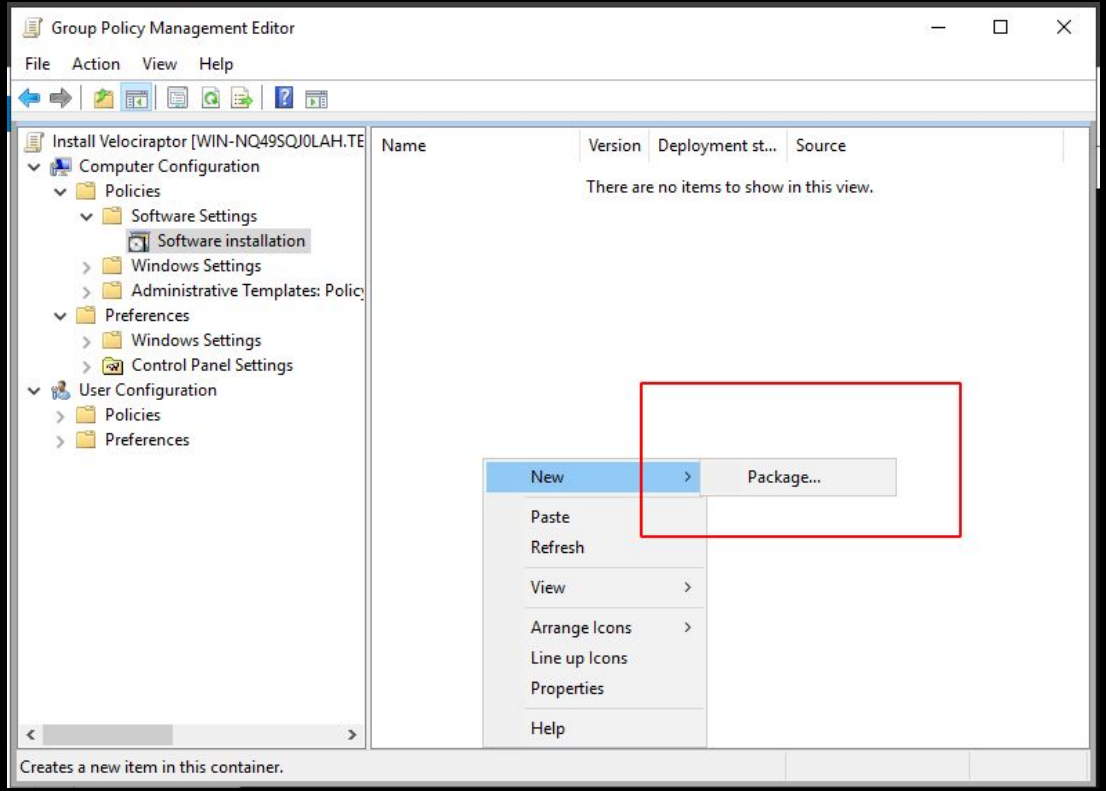
Ensure the new scheduled task is run as system



Using scheduled tasks you can run any binary - use this method to run interactive collection if you do not have a dedicated Velociraptor server

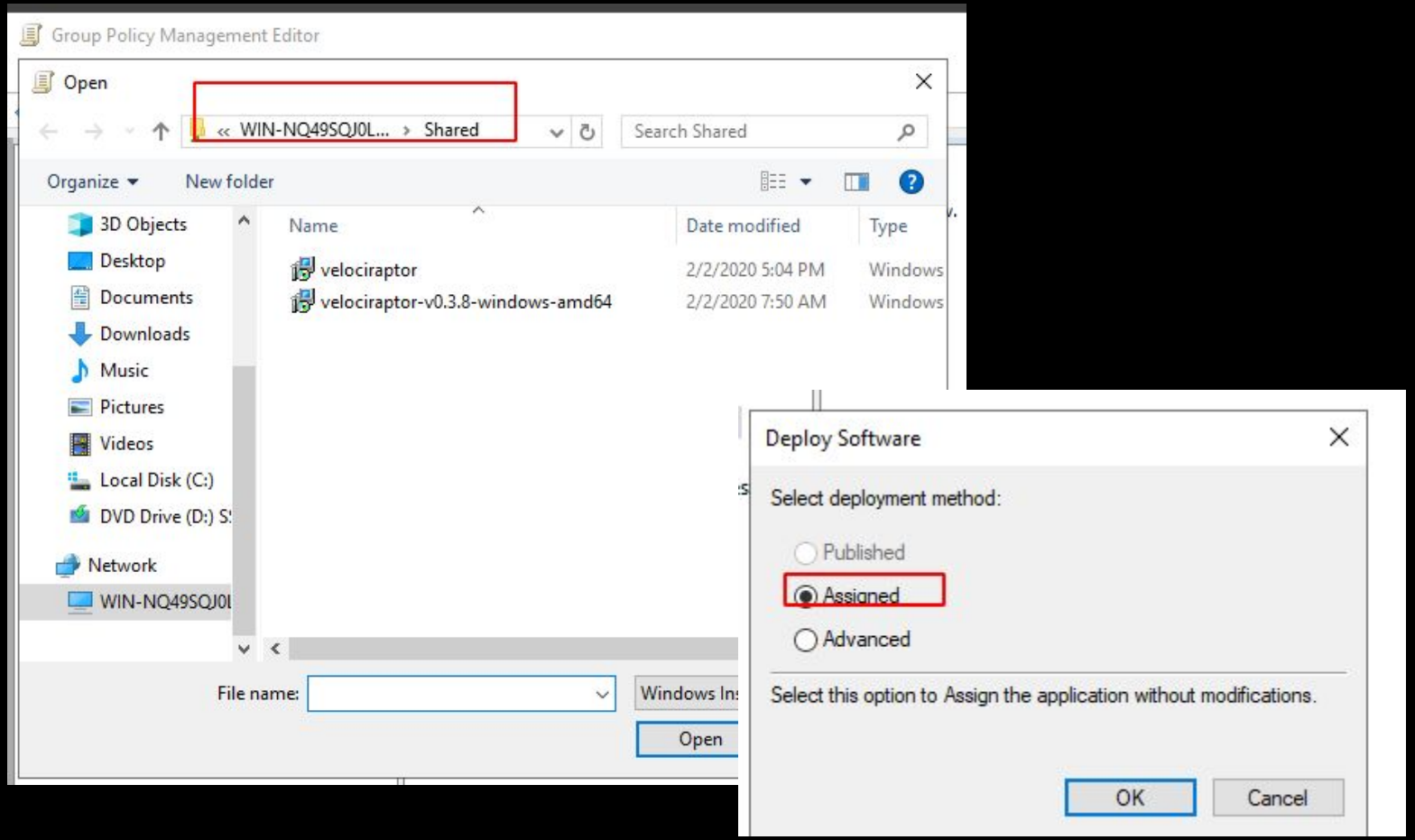


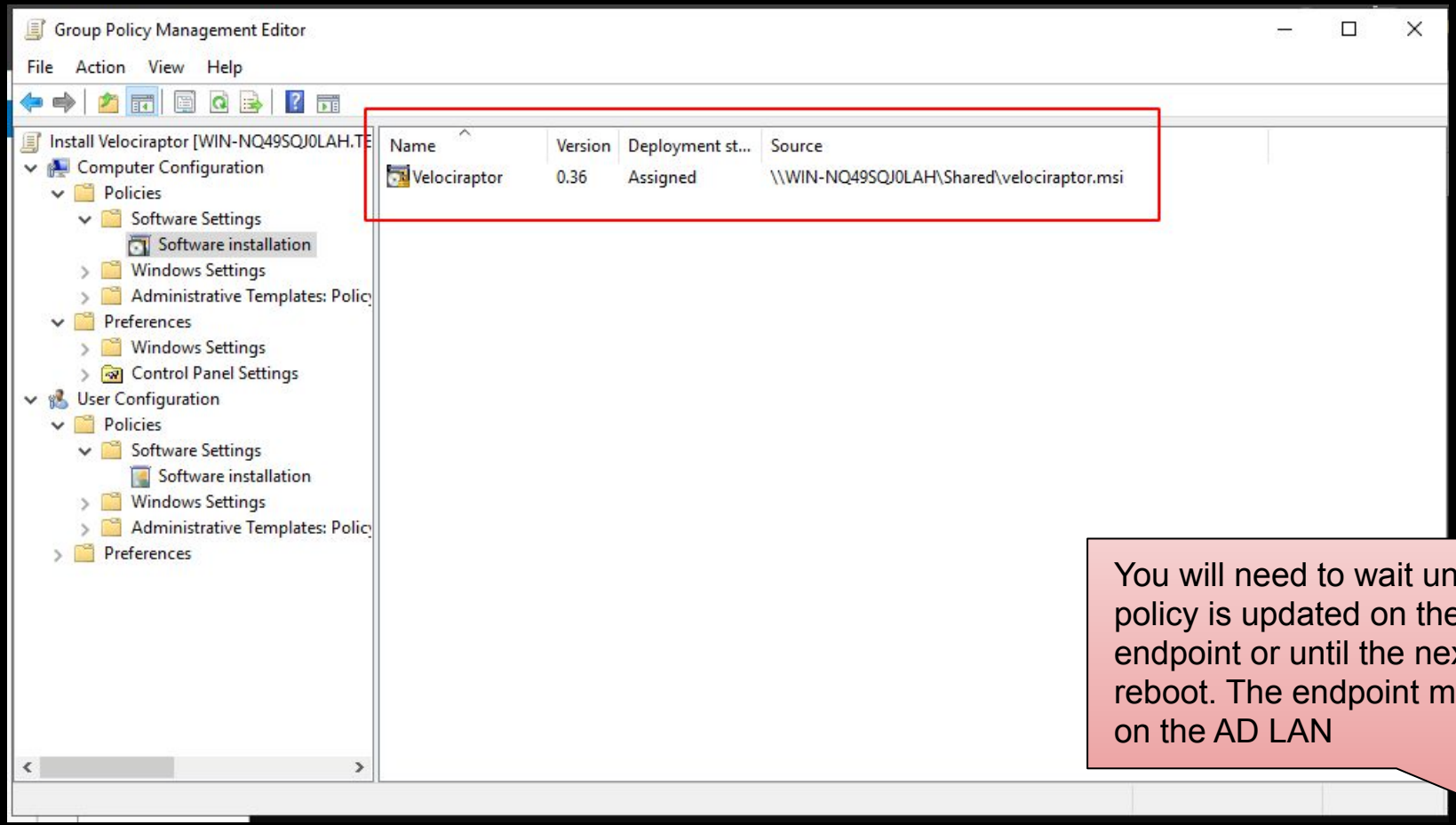
Ensure the new scheduled task is run only once



Method 2 - install via assigned software packages in GPO

The main advantage here is that it is possible to upgrade or uninstall Velociraptor easily





You will need to wait until group policy is updated on the endpoint or until the next reboot. The endpoint must be on the AD LAN

A Velociraptor GUI tour

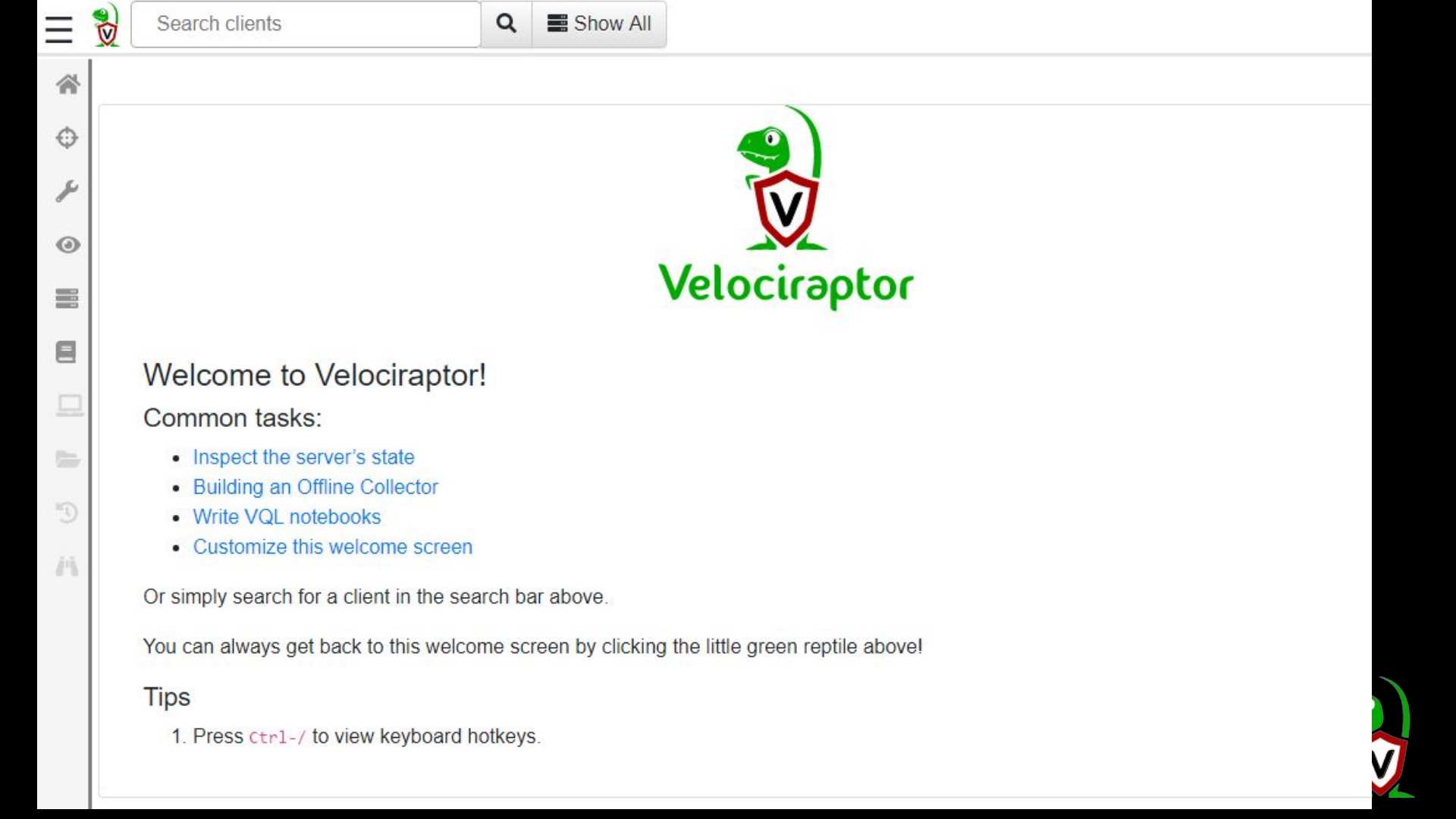
The Dashboard

The **Dashboard** shows the current state of the installation:

- How many clients are connected
- Current CPU load and memory footprint on the server.

When running hunts or intensive processing, memory and CPU requirements will increase but not too much.

You can customize the dashboard - it's also just an artifact.


Search clients
Show All
Velociraptor
Welcome to Velociraptor!
Common tasks:
Inspect the server's state
Building an Offline Collector
Write VQL notebooks
Customize this welcome screen
Or simply search for a client in the search bar above.
You can always get back to this welcome screen by clicking the little green reptile above!
Tips
1. Press Ctrl-/ to view keyboard hotkeys.

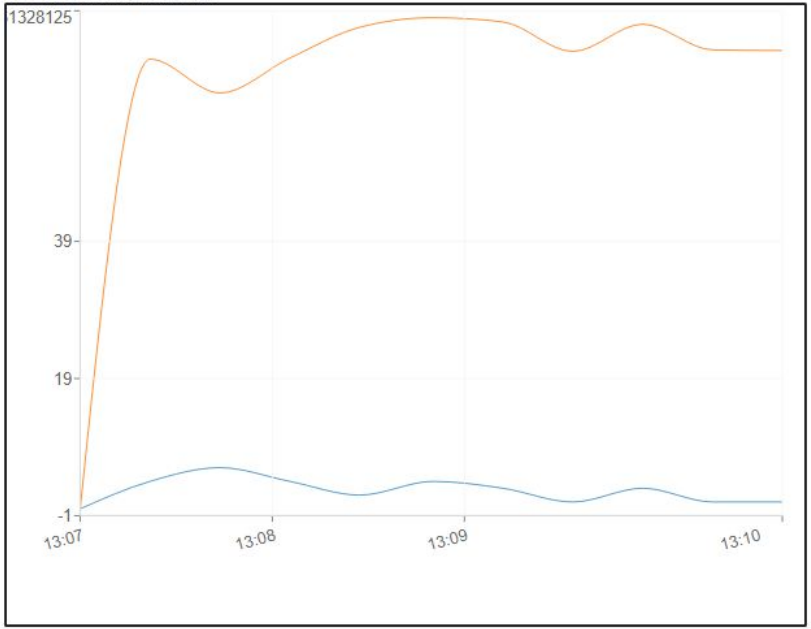
Search clients | Show All | admin

Last Day

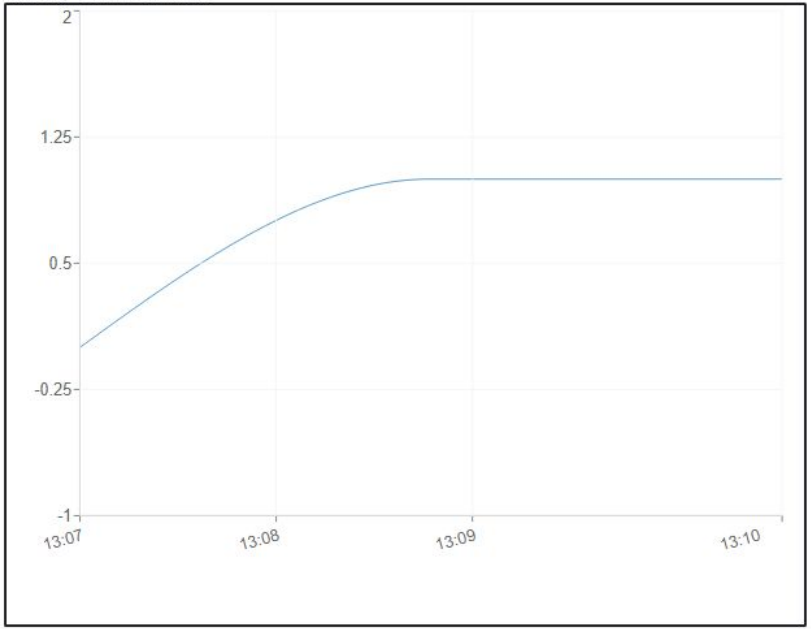
Server status

The following are total across all frontends.

CPU and Memory Utilization



Currently Connected Clients



Clients have a persistent connection to the server.

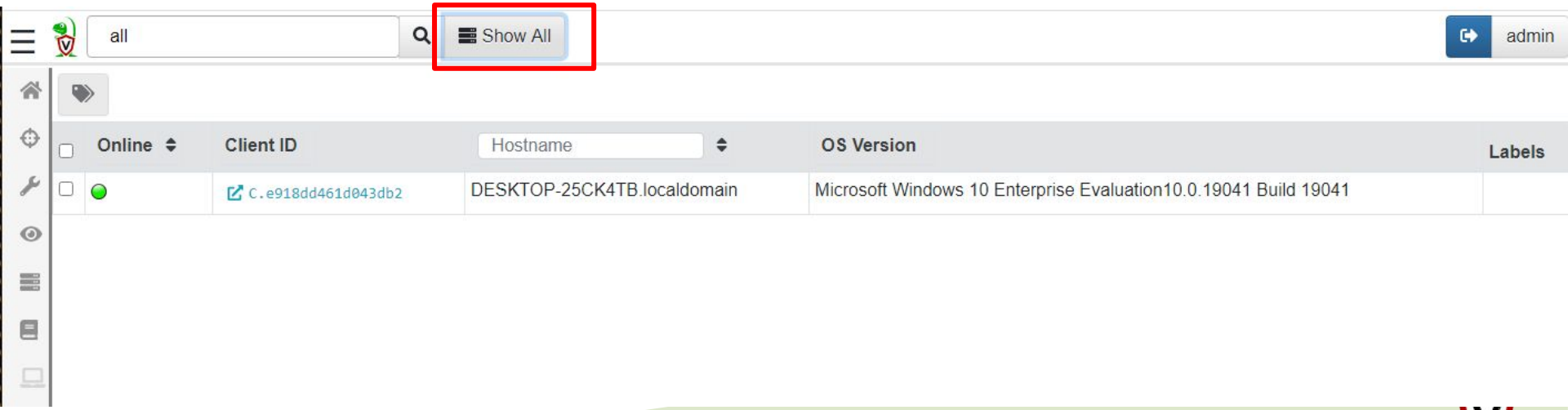
They're ready to receive your commands.

Interactively investigate individual clients

Searching for a client

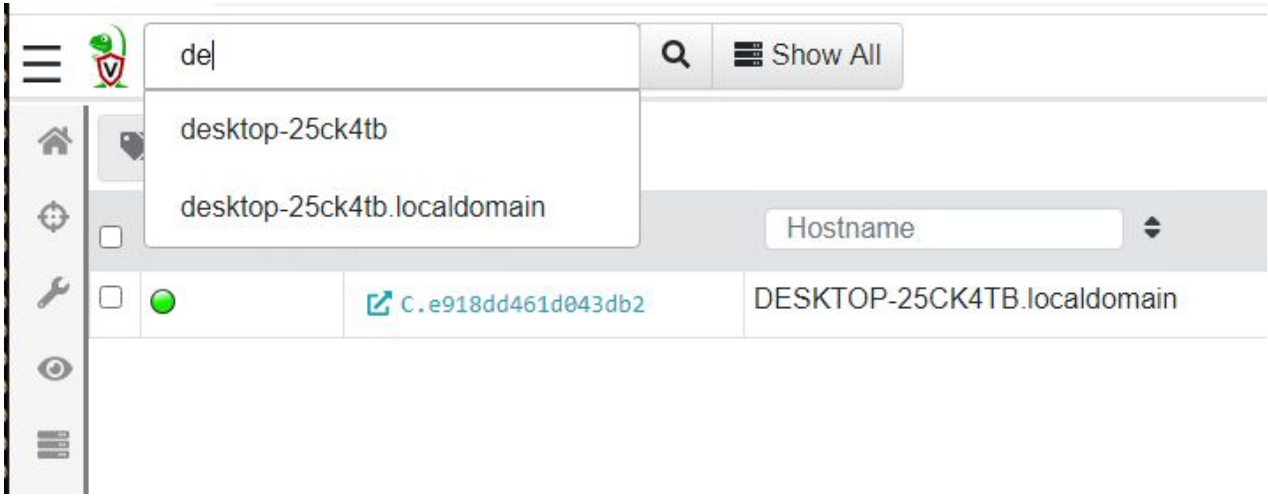
To work with a specific client we need to search for it.

Press the **Search** or **Show All** icon to see some clients



Search for clients hostname, label, or client ID.

You can start typing the hostname to auto-complete



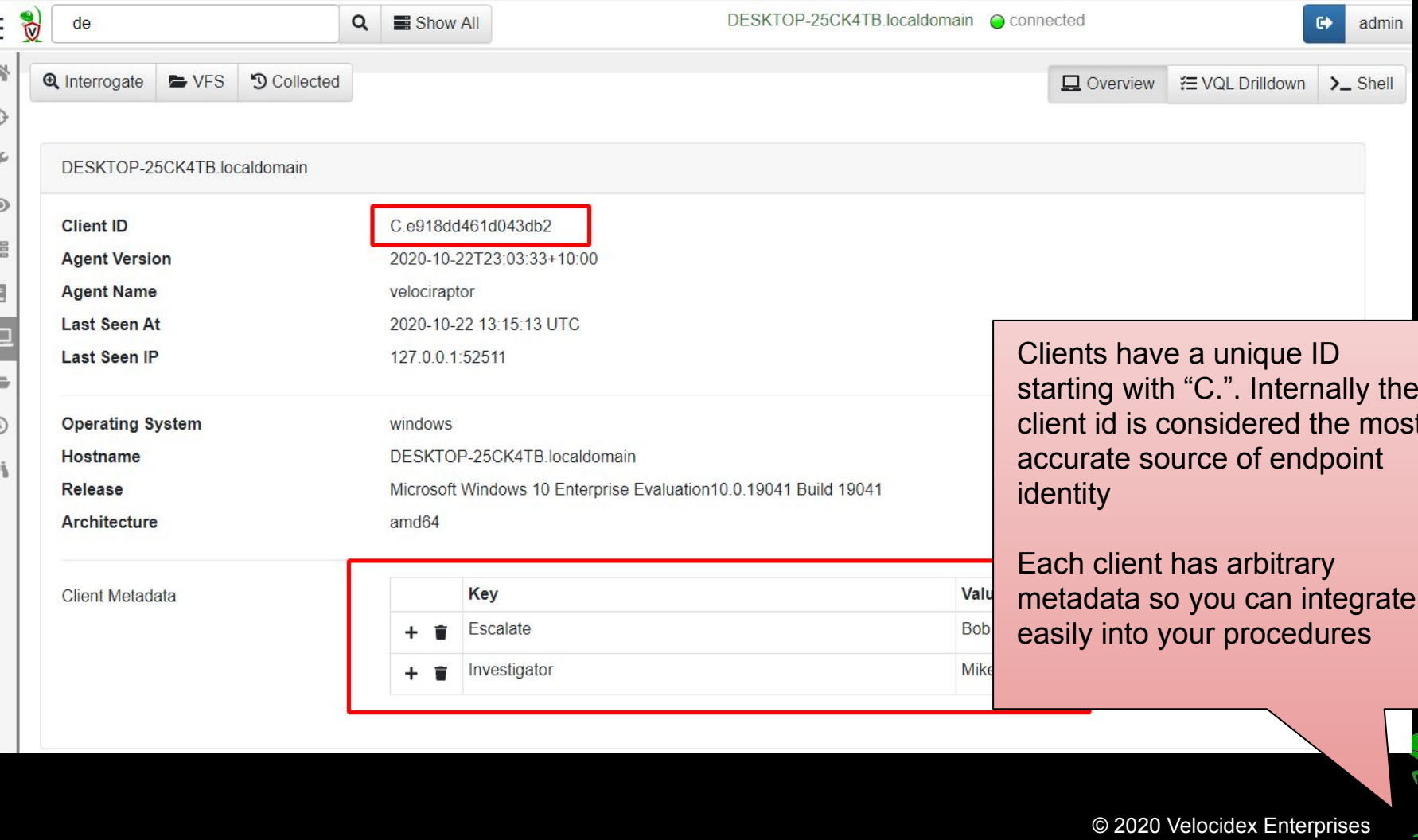
Client overview

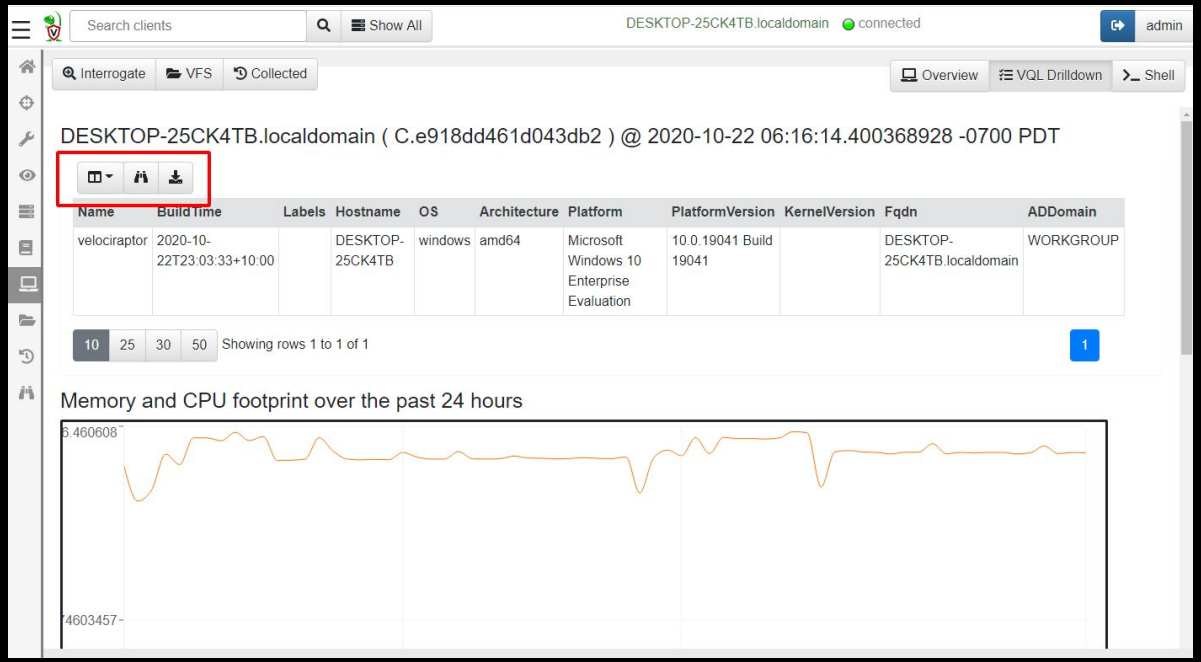
The server collects some high level information about each endpoint.

Click **VQL Drilldown** to see more detailed information:

- ❑ Client version
- ❑ Client footprint (memory and CPU)

You can customize the information collected and shown by editing the Generic.Client.Info artifact.

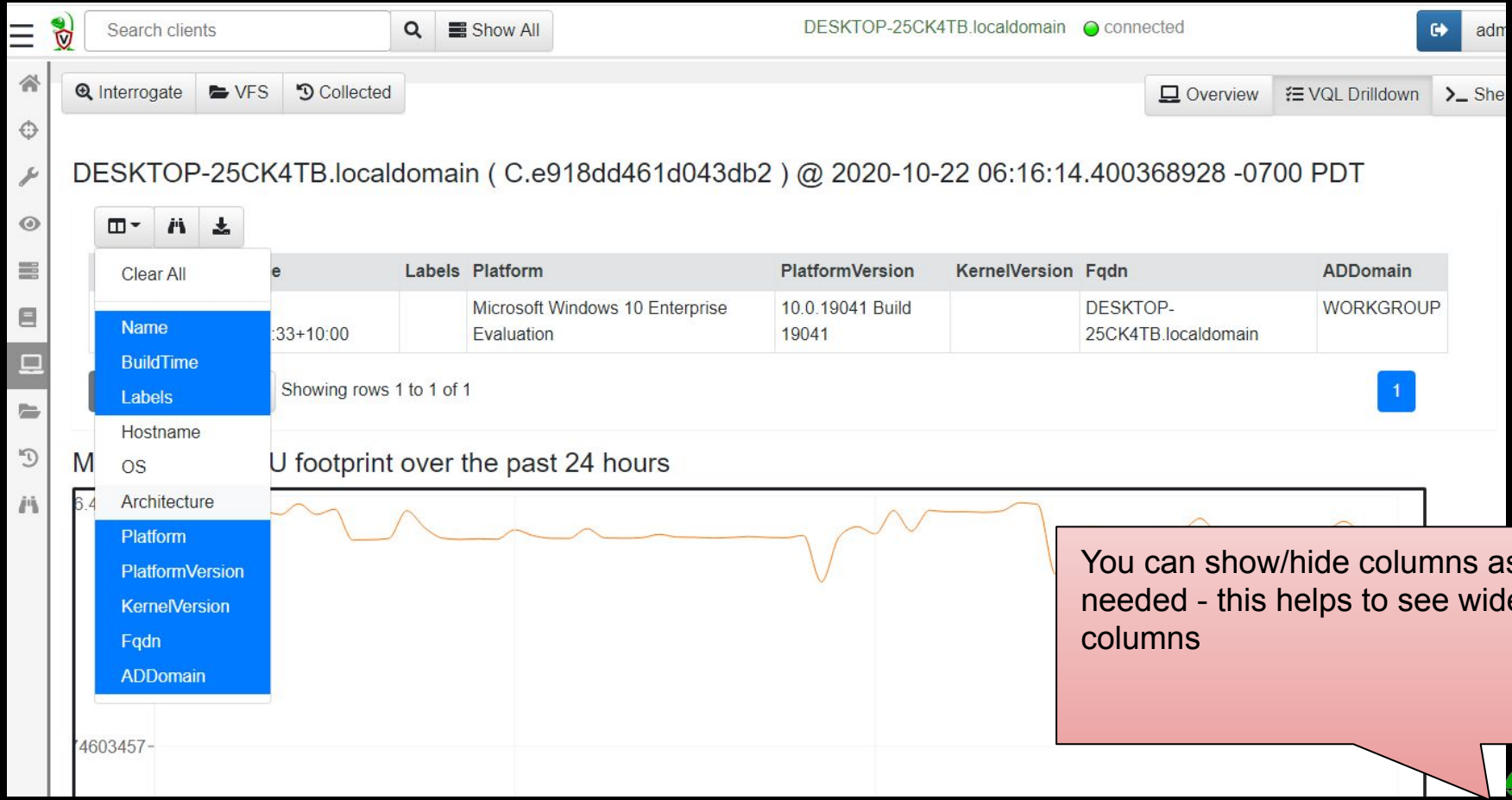


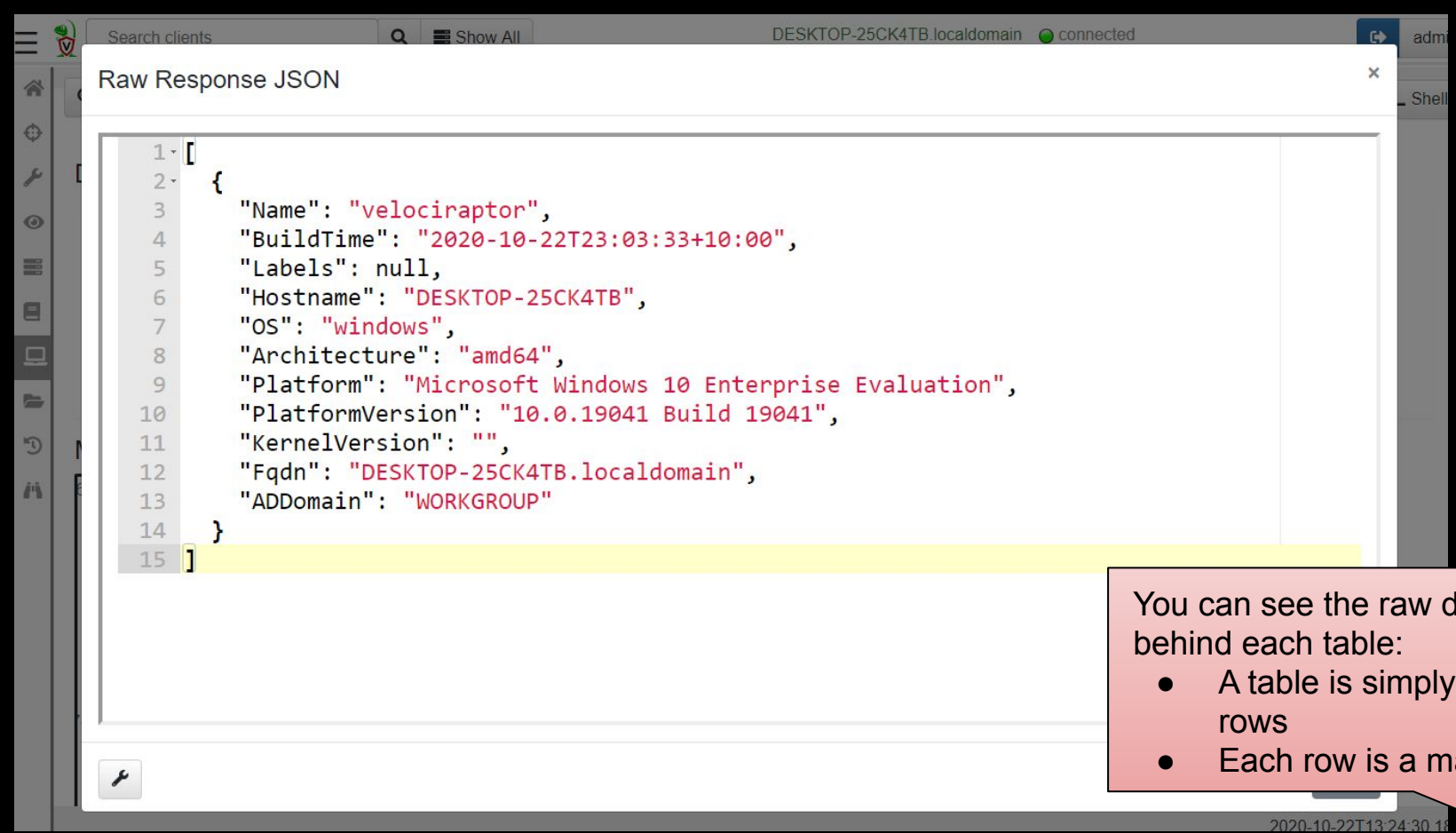


By default, VQL Drill Down shows the recent memory and CPU load of Velociraptor on the endpoint as well as the list of users.

The GUI consists of familiar widgets: Here we can see the table widget which repeats often

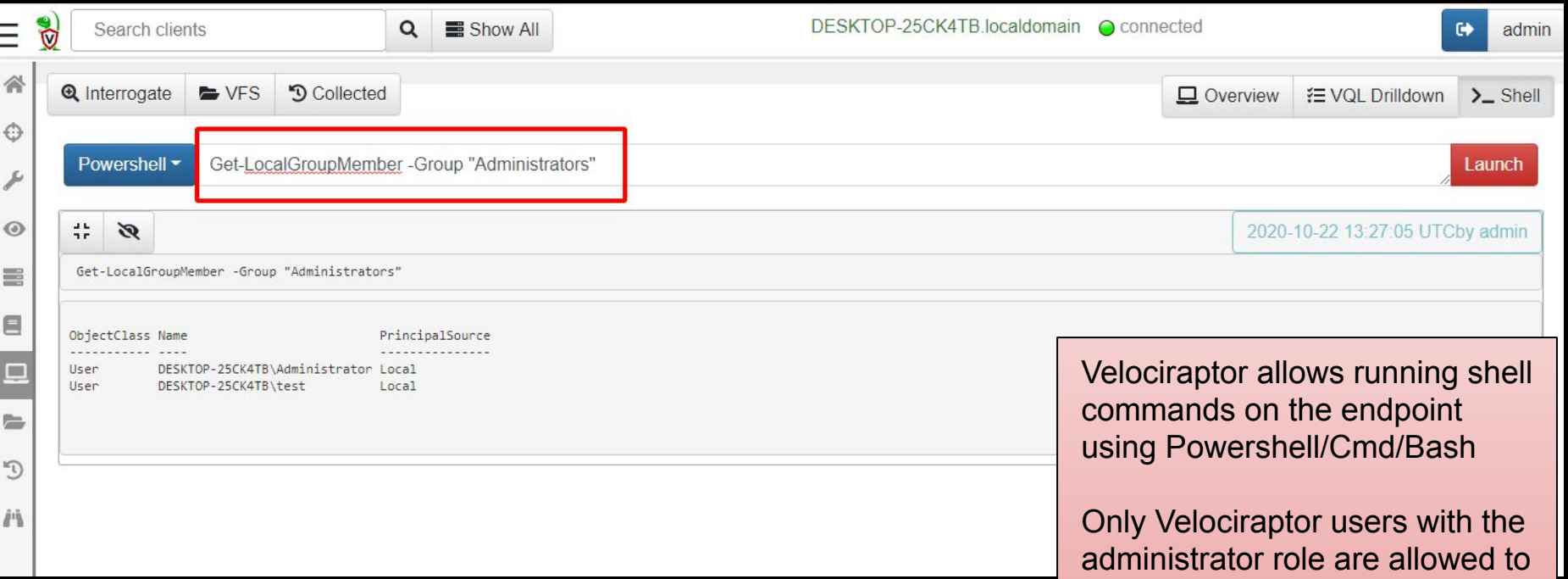
This screen simply shows the report of the Generic.Client.Info artifact - you can edit the artifact to collect more/different info.





You can see the raw data behind each table:

- A table is simply a list of rows
- Each row is a mapping



You can disable client shell ability by configuration policy - but this limits your DFIR efficacy.

Velociraptor allows running shell commands on the endpoint using Powershell/Cmd/Bash

Only Velociraptor users with the administrator role are allowed to do this!

Actions are logged and audited

Interactively fetching files from the endpoint

The Virtual File System (VFS)

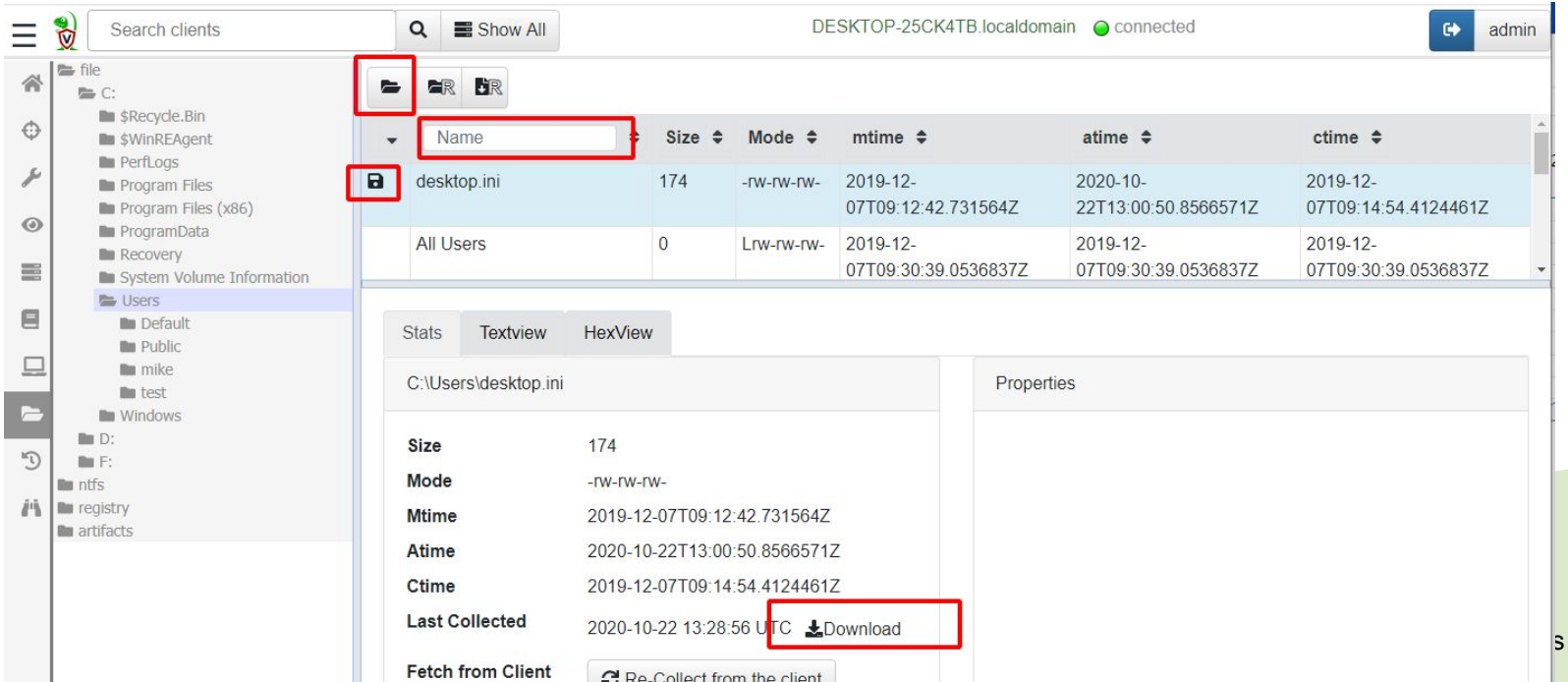
The VFS visualizes some server-side information we collect about the clients.

Top level corresponds to the type of information we collect:

- **File** - Access the file system using the filesystem API
- **NTFS** - Access the file system using raw NTFS parsing (Windows Only)
- **Registry** - Access the Windows Registry using the Registry API (Windows Only)
- **Artifacts** - A view of all artifacts collected from the client sorted by artifact type, and then times when they were collected.

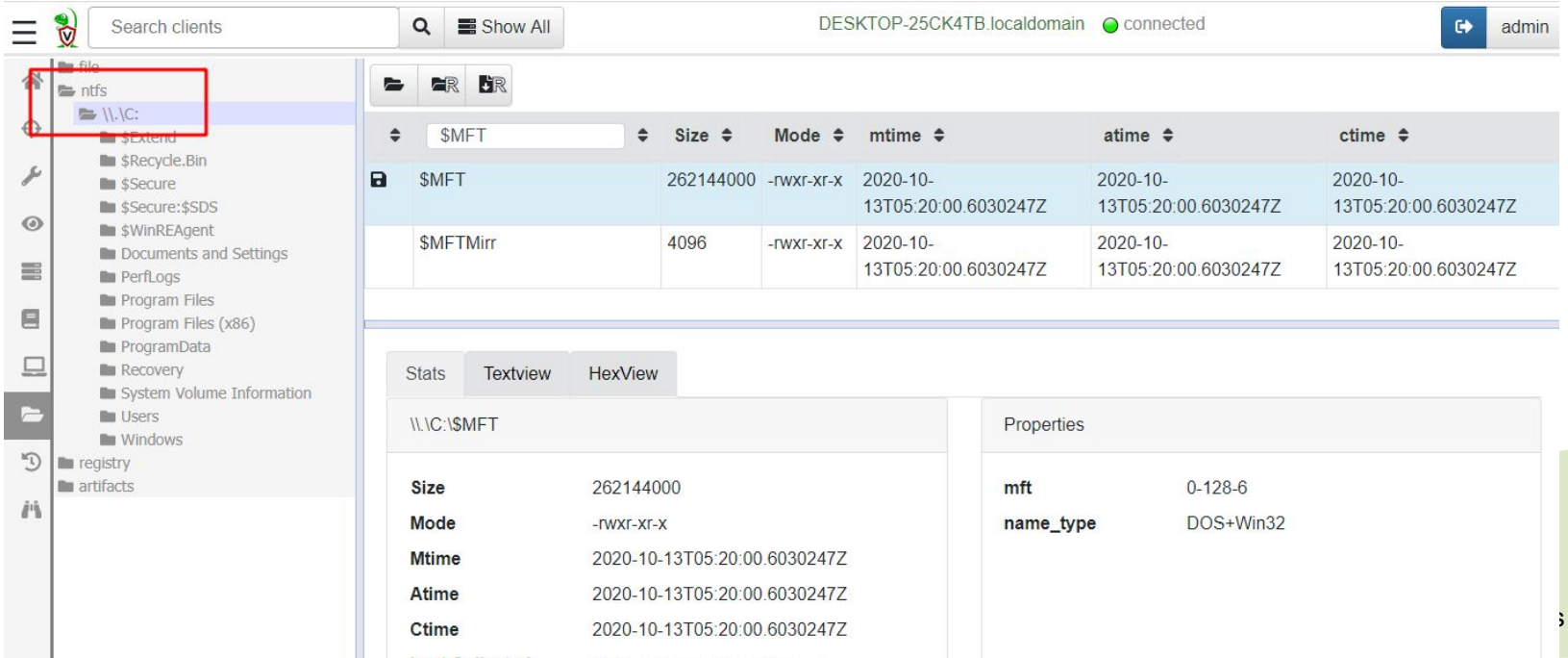
File accessor

Uses the OS APIs to access files (unless locked then it fallback to NTFS)



NTFS Accessor

Uses raw NTFS parsing providing access to special files and ADS



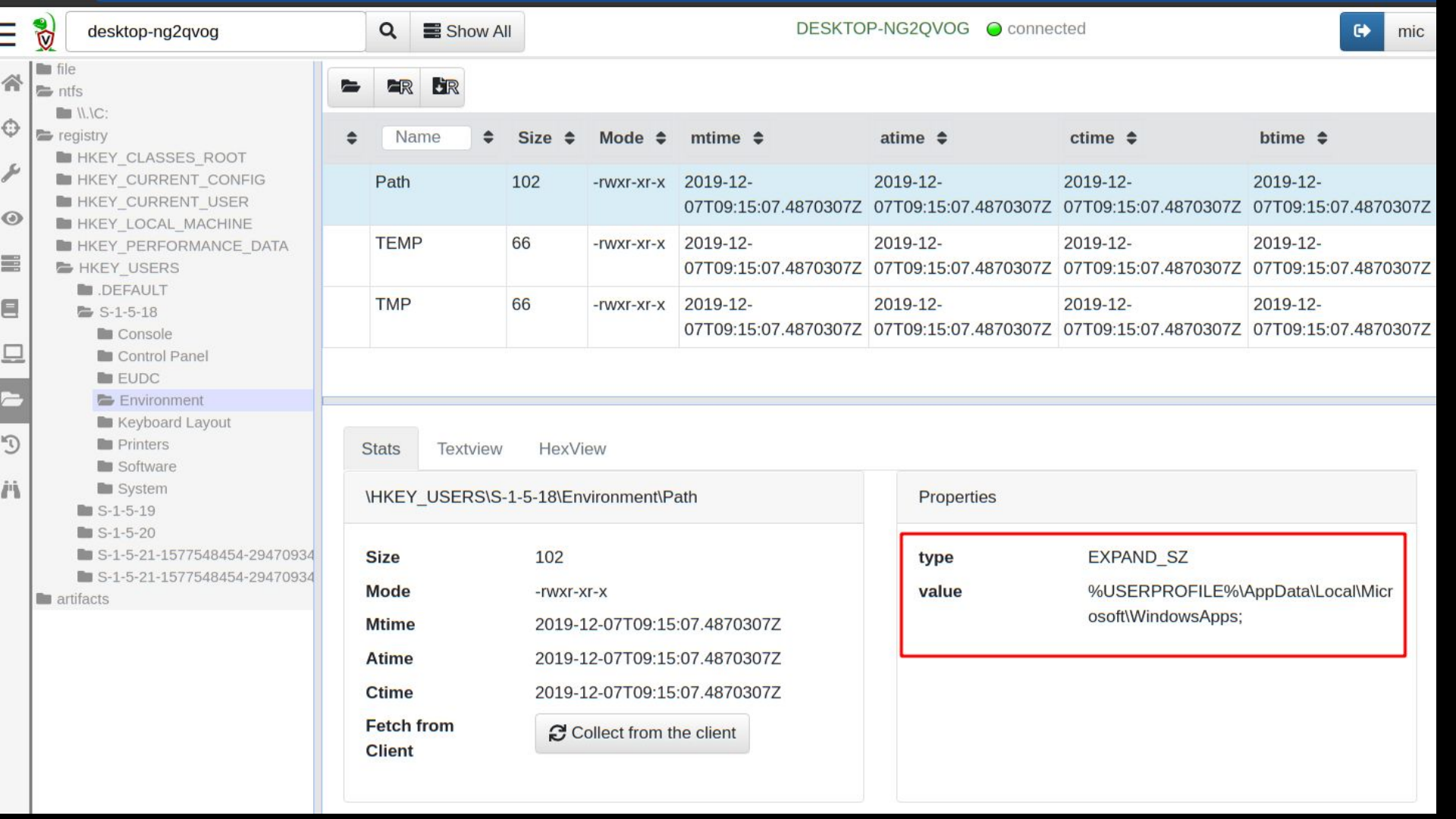
Registry Accessor

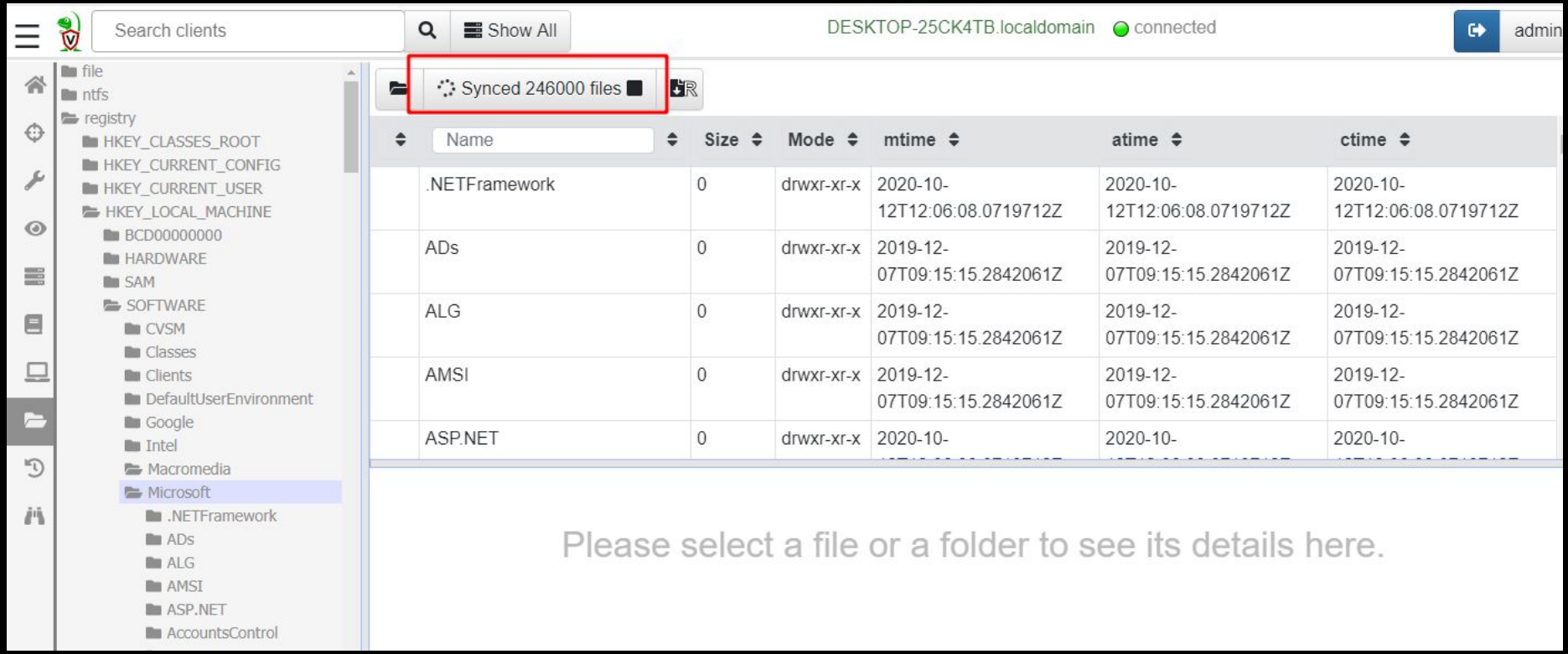
Provides access to registry using the Windows API.

Keys are like directories and Values are files.

Since Values are typically small, they are also retrieved as a result of a directory listing - in most cases there is no need to download content explicitly.

Note that registry mapping occurs so take care when accessing virtual keys like HKEY_CURRENT_USER or HKEY_USERS

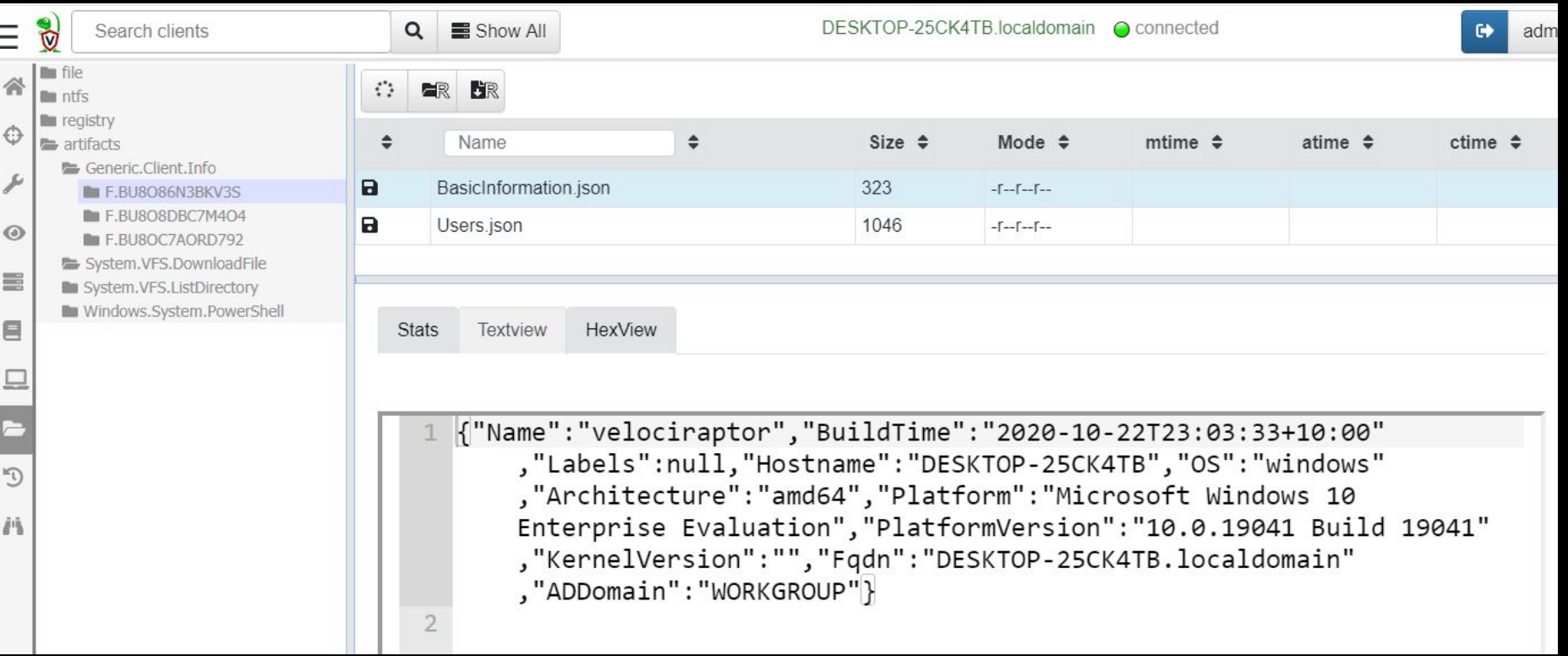

desktop-ng2qvog
Show All
DESKTOP-NG2QVOG
connected
mic
file
ntfs
\\.\C:
registry
HKEY_CLASSES_ROOT
HKEY_CURRENT_CONFIG
HKEY_CURRENT_USER
HKEY_LOCAL_MACHINE
HKEY_PERFORMANCE_DATA
HKEY_USERS
.DEFAULT
S-1-5-18
Console
Control Panel
EUDC
Environment
Keyboard Layout
Printers
Software
System
S-1-5-19
S-1-5-20
S-1-5-21-1577548454-29470934
S-1-5-21-1577548454-29470934
artifacts
Name
Size
Mode
mtime
atime
ctime
btime
Path 102 -rwxr-xr-x 2019-12-07T09:15:07.4870307Z 2019-12-07T09:15:07.4870307Z 2019-12-07T09:15:07.4870307Z 2019-12-07T09:15:07.4870307Z
TEMP 66 -rwxr-xr-x 2019-12-07T09:15:07.4870307Z 2019-12-07T09:15:07.4870307Z 2019-12-07T09:15:07.4870307Z 2019-12-07T09:15:07.4870307Z
TMP 66 -rwxr-xr-x 2019-12-07T09:15:07.4870307Z 2019-12-07T09:15:07.4870307Z 2019-12-07T09:15:07.4870307Z 2019-12-07T09:15:07.4870307Z
Stats
Textview
HexView
\HKEY_USERS\S-1-5-18\Environment\Path
Size 102
Mode -rwxr-xr-x
Mtime 2019-12-07T09:15:07.4870307Z
Atime 2019-12-07T09:15:07.4870307Z
Ctime 2019-12-07T09:15:07.4870307Z
Fetch from Client
Collect from the client
Properties
type EXPAND_SZ
value %USERPROFILE%\AppData\Local\Microsoft\WindowsApps;



Artifacts accessor

This shows the artifacts collected from the endpoint grouped by artifact

This is useful to see the timeline of the same artifact collected at different times.

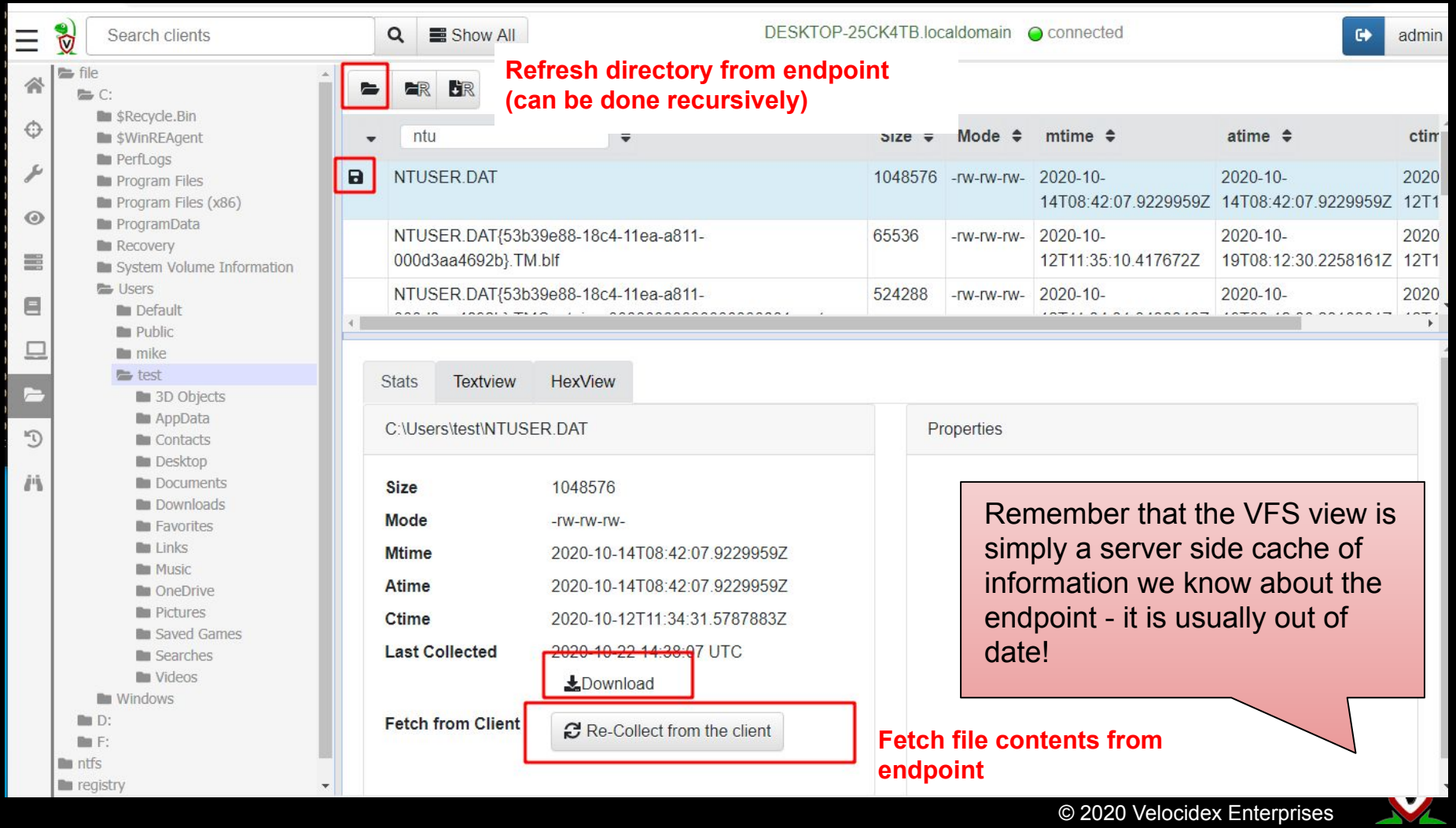


Navigating the interface

Click the “Refresh this directory” will schedule a directory listing artifact and wait for the results (usually very quick if the endpoint is online).

The “Recursively refresh this directory” will schedule a recursive refresh - this may take some time! After this operation a lot of the VFS will be pre-populated already.

“Collect from client” will retrieve the file data to the server. After which, the floppy disk sign indicates that we have file data available and you can click the “Download” link to get a copy of the file.

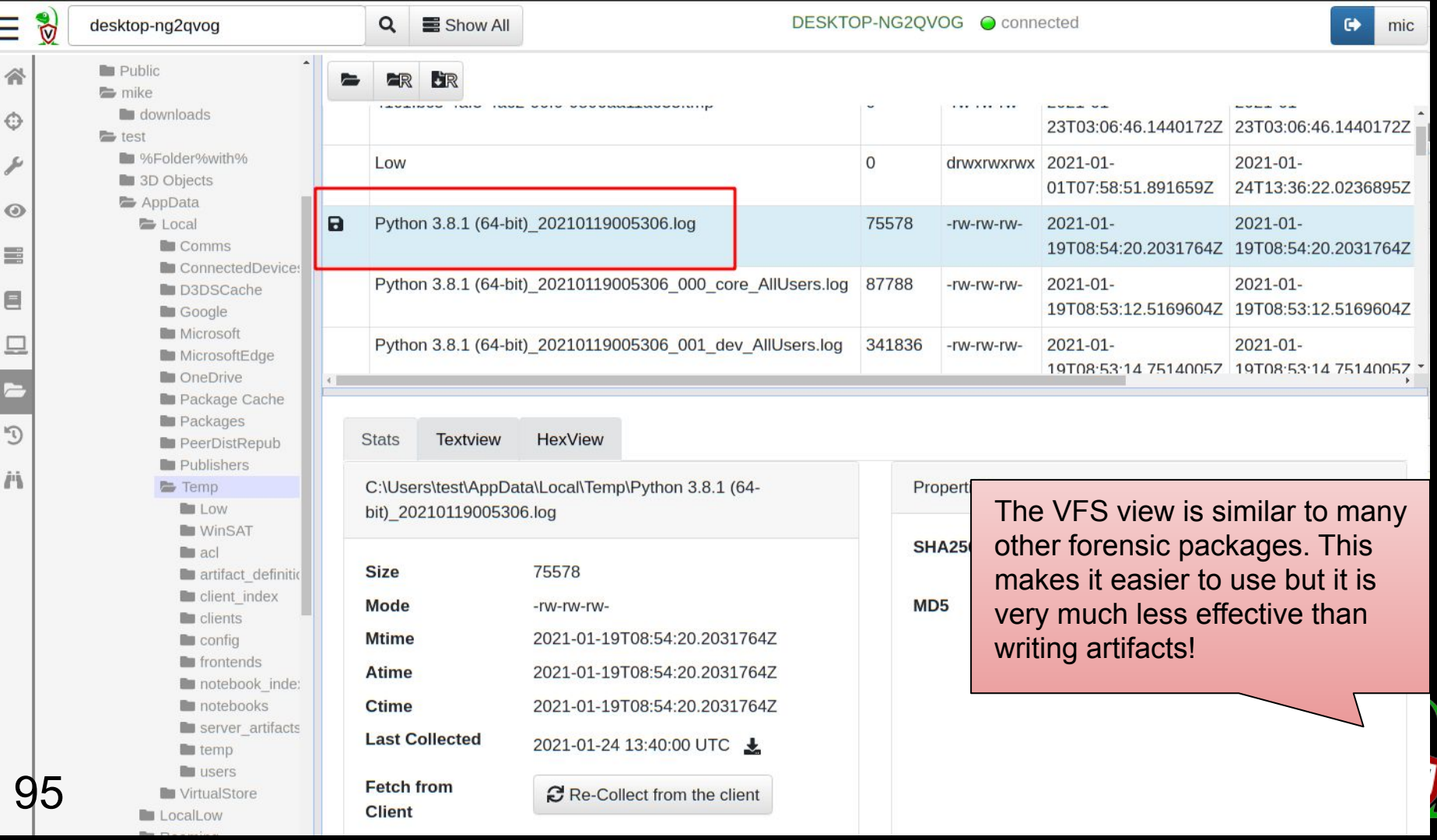


Exercise: Determine user activity

Task: We suspect a user account had been compromised.

Did the user download malware?

- ❏ Freely explore the interface to answer this question
- ❏ Useful artifacts include
  - ❏ Download directory content
  - ❏ Internet browser history
  - ❏ Temporary files


desktop-ng2qvog
Show All
DESKTOP-NG2QVOG
connected
mic
Public
mike
downloads
test
%Folder%with%
3D Objects
AppData
Local
Comms
ConnectedDevices
D3DSCache
Google
Microsoft
MicrosoftEdge
OneDrive
Package Cache
Packages
PeerDistRepub
Publishers
Temp
Low
WinSAT
acl
artifact_definitic
client_index
clients
config
frontends
notebook_inde:
notebooks
server_artifacts
temp
users
VirtualStore
LocalLow
23T03:06:46.1440172Z
23T03:06:46.1440172Z
Low
0
drwxrwxrwx
2021-01-01T07:58:51.891659Z
2021-01-24T13:36:22.0236895Z
Python 3.8.1 (64-bit)_20210119005306.log
75578
-rw-rw-rw-
2021-01-19T08:54:20.2031764Z
2021-01-19T08:54:20.2031764Z
Python 3.8.1 (64-bit)_20210119005306_000_core_AllUsers.log
87788
-rw-rw-rw-
2021-01-19T08:53:12.5169604Z
2021-01-19T08:53:12.5169604Z
Python 3.8.1 (64-bit)_20210119005306_001_dev_AllUsers.log
341836
-rw-rw-rw-
2021-01-19T08:53:14.7514005Z
2021-01-19T08:53:14.7514005Z
Stats
Textview
HexView
C:\Users\test\AppData\Local\Temp\Python 3.8.1 (64-bit)_20210119005306.log
Size
75578
Mode
-rw-rw-rw-
Mtime
2021-01-19T08:54:20.2031764Z
Atime
2021-01-19T08:54:20.2031764Z
Ctime
2021-01-19T08:54:20.2031764Z
Last Collected
2021-01-24 13:40:00 UTC
Fetch from Client
Re-Collect from the client
SHA25
MD5
The VFS view is similar to many other forensic packages. This makes it easier to use but it is very much less effective than writing artifacts!

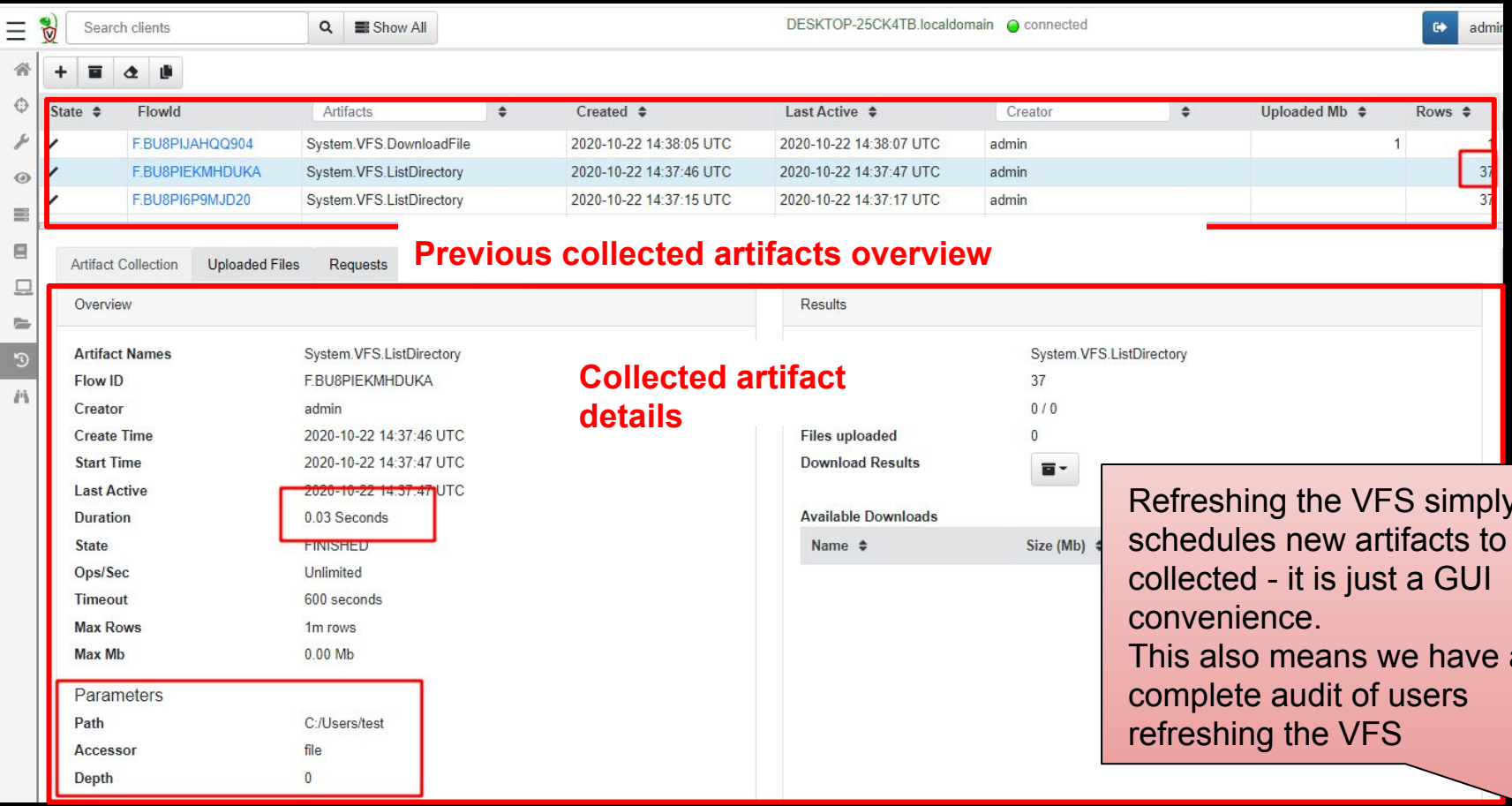
Velociraptor Artifacts
Fast, Efficient, Surgical

Velociraptor artifacts

Velociraptor is just a VQL engine!

We package VQL queries in Artifacts:

- ❏ YAML files
- ❏ Include human description
- ❏ Package related VQL queries into “Sources”
- ❏ Take parameters for customization
- ❏ Can in turn be used in VQL as well...



Refreshing the VFS simply schedules new artifacts to be collected - it is just a GUI convenience.
This also means we have a complete audit of users refreshing the VFS

Velociraptor uses expert knowledge to find the evidence

A key objective of Velociraptor is encapsulating DFIR knowledge into the platform, so you don't need to be a DFIR expert.

- ❑ We have high level questions to answer
- ❑ We know where to look for evidence of user / system activities

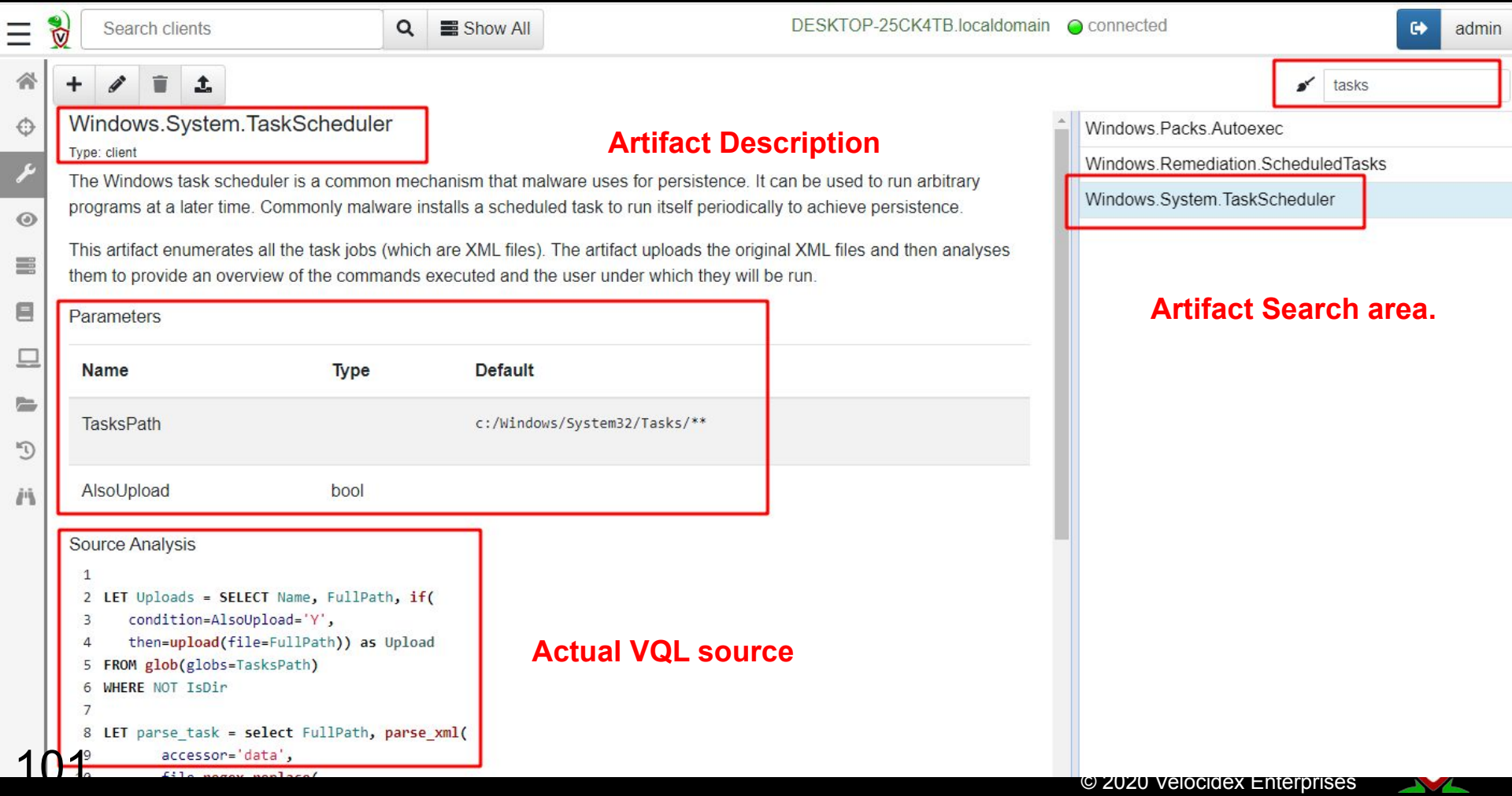
We build artifacts to collect and analyze the evidence in order to answer our investigative questions.

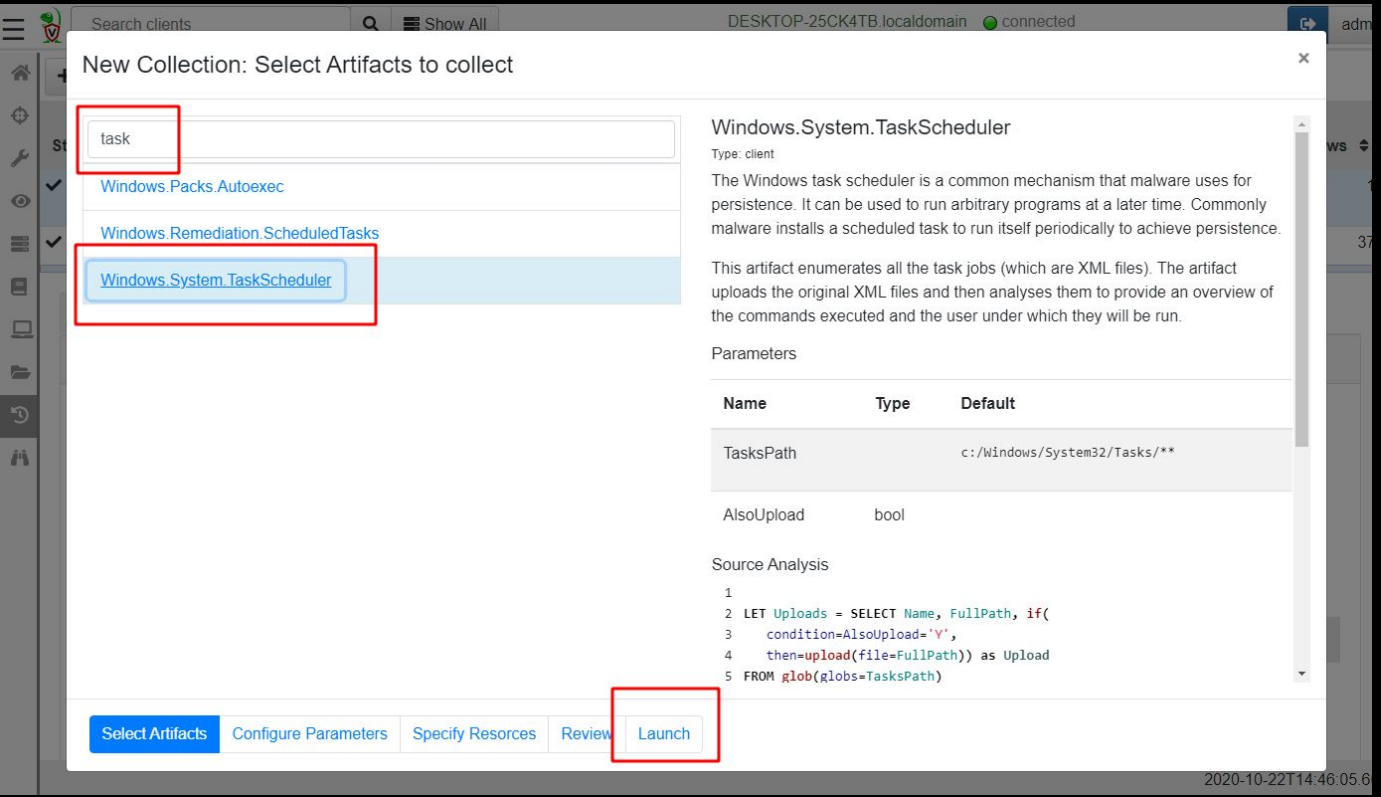
Velociraptor's superpower: user specified artifacts

An artifact is a YAML file …

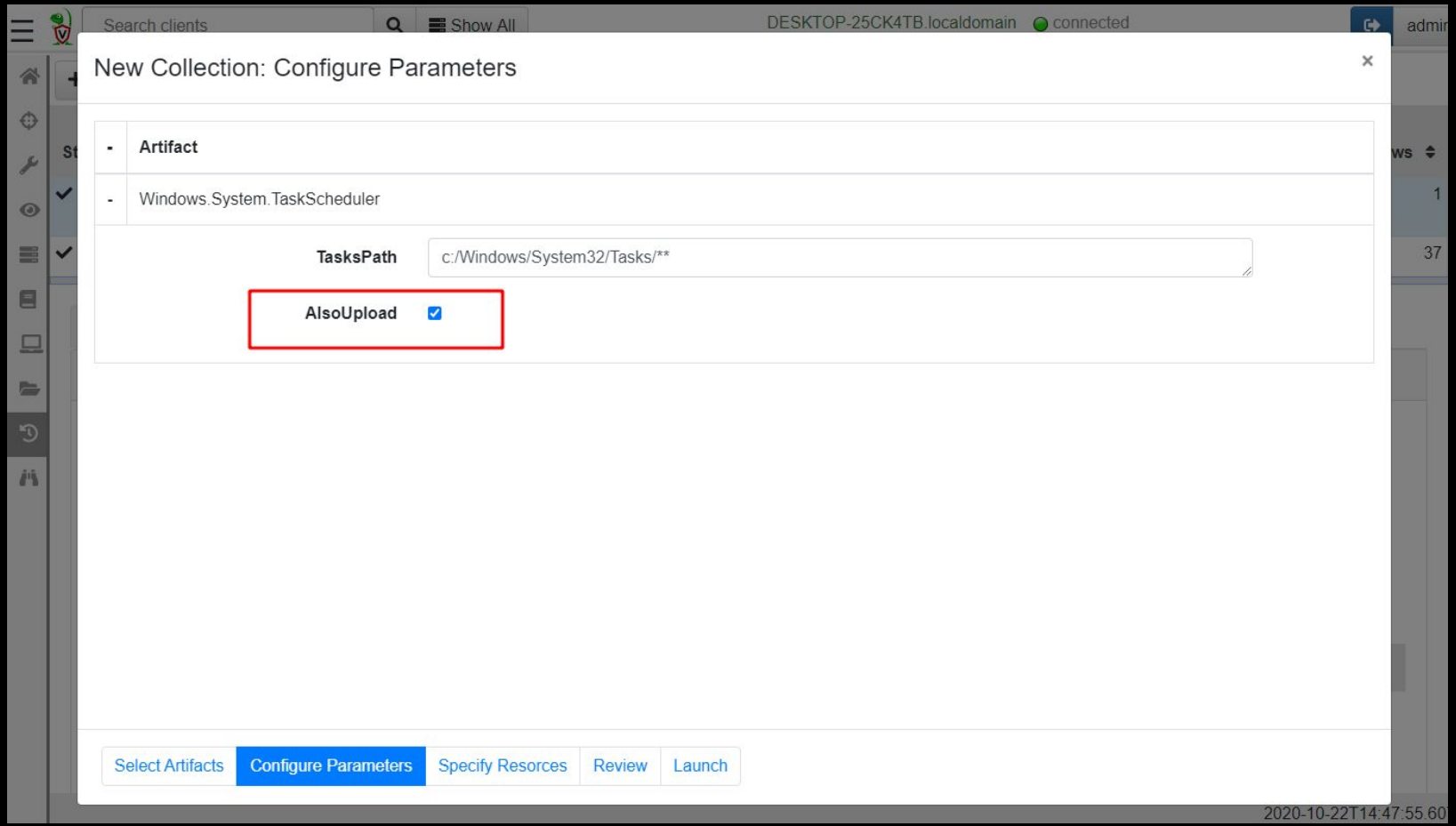
- ❏ (therefore user-readable, shareable and editable)
- ❏ … that answers a question …
- ❏ … by collecting data from the endpoint …
- ❏ … and reporting on this data in a human readable way.

Artifacts encode expert knowledge into human reusable components.





To collect a new artifact, from the *Collected Artifacts* screen, click *Collect new artifact* and search for it. Select *Add* to add it to this collection. When finished simply click *Next*.



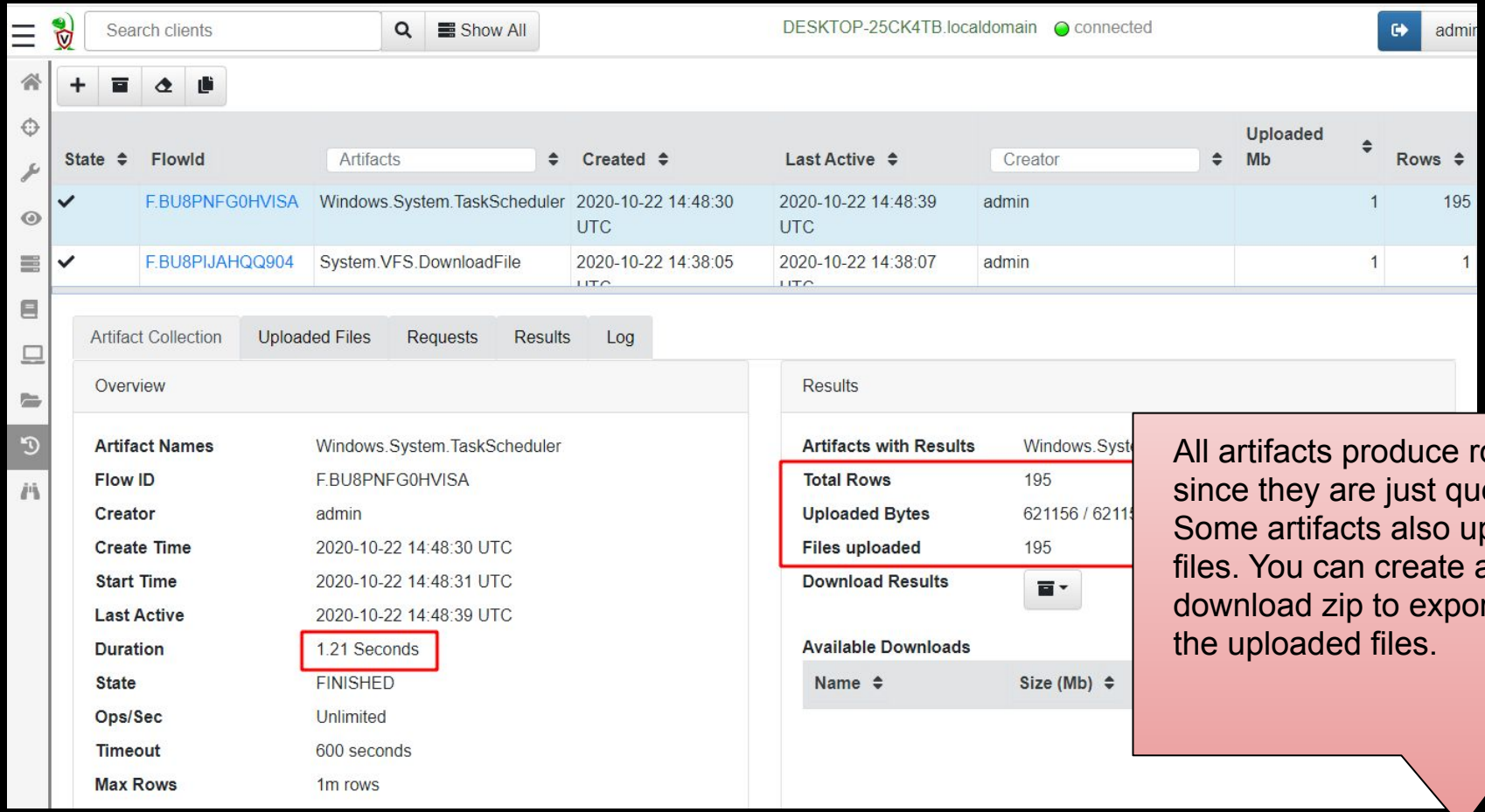
Velociraptor Artifacts

Velociraptor comes with a large number of artifact types

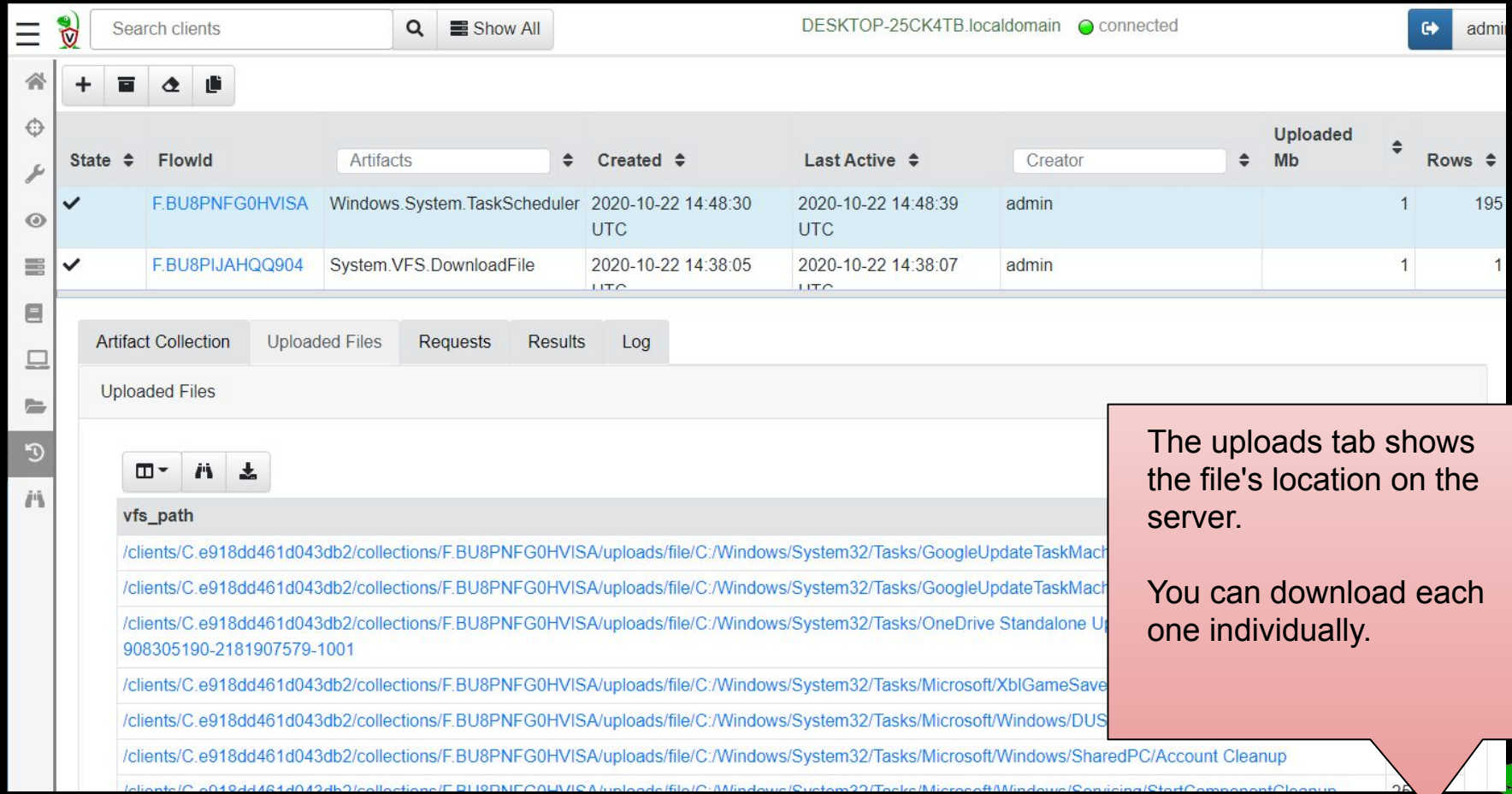
1. Client Artifacts run on the endpoint
2. Client Event artifacts monitor the endpoint
3. Server Artifacts run on the server
4. Server Event artifacts monitor for events on the server.

Depending on context, the GUI artifact search screen will only show the relevant artifact types.

The **View Artifacts** page shows all types as well as details about each one.

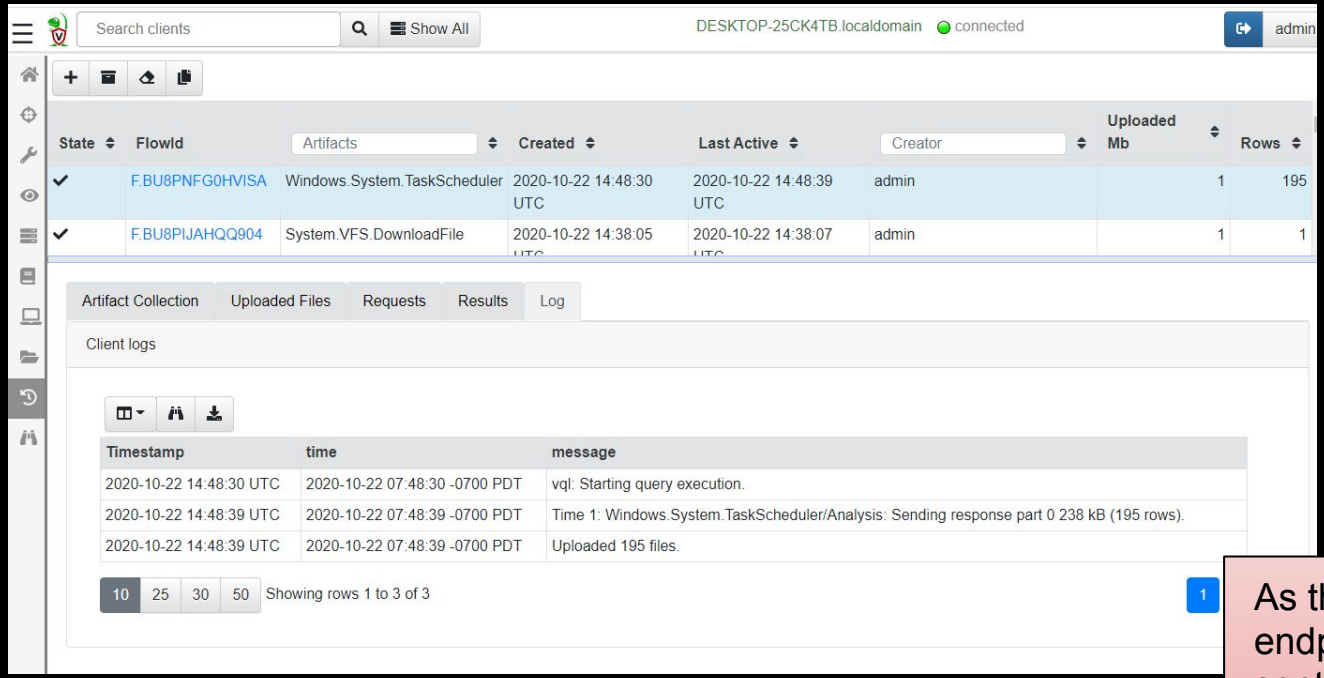


All artifacts produce rows since they are just queries. Some artifacts also upload files. You can create a download zip to export all the uploaded files.

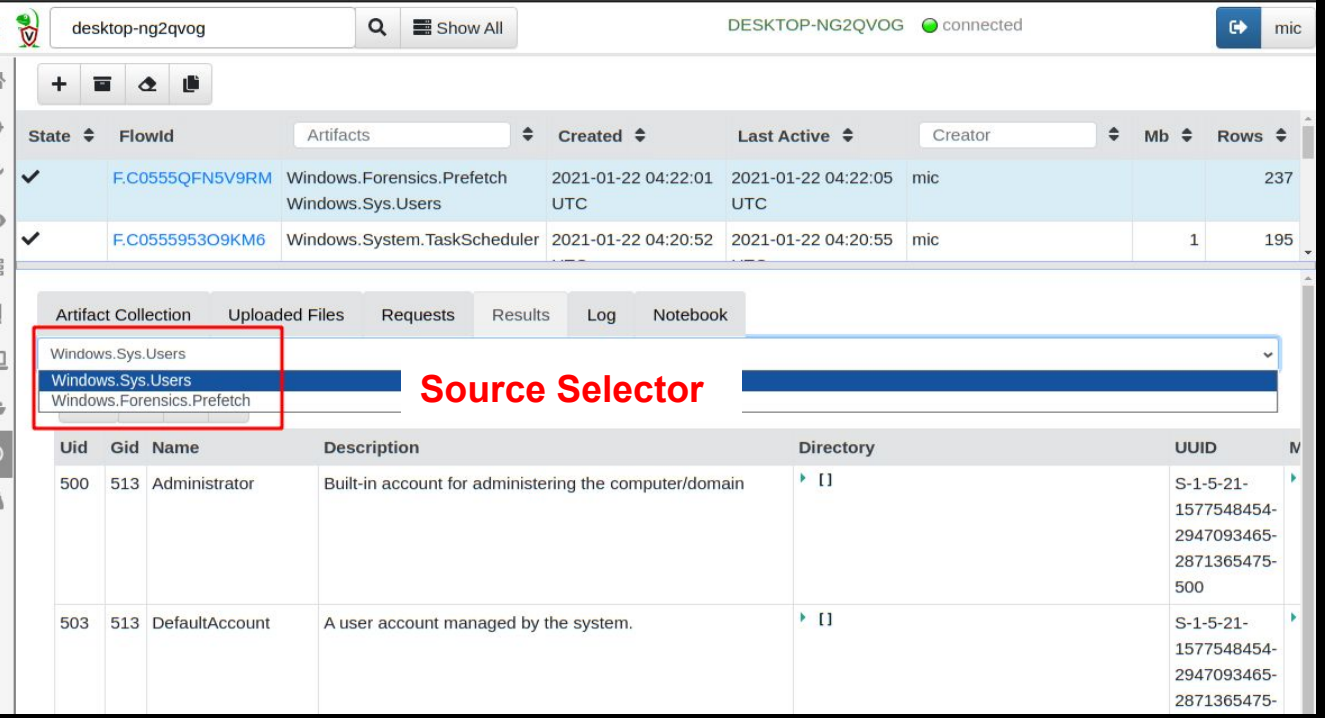


The uploads tab shows the file's location on the server.

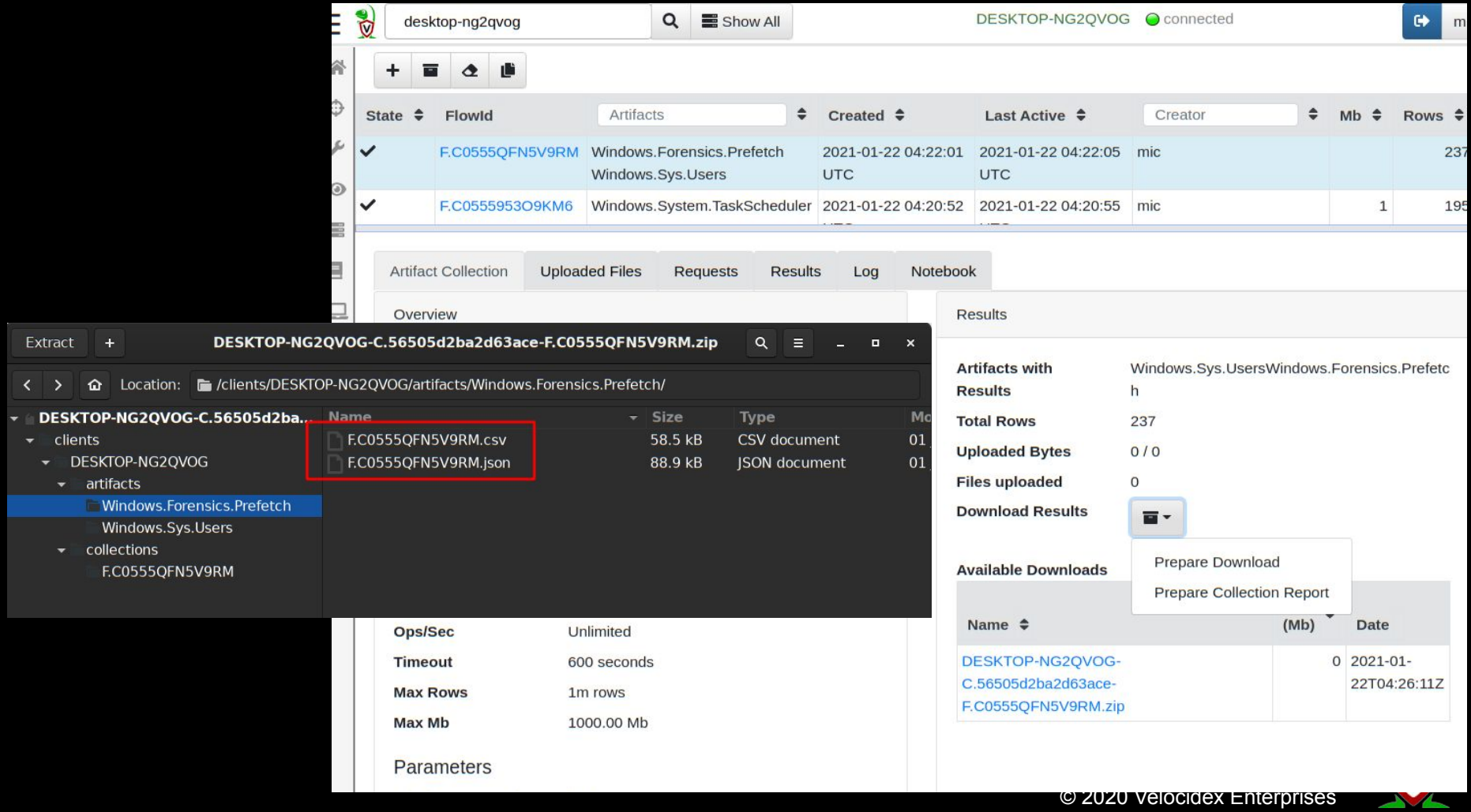
You can download each one individually.

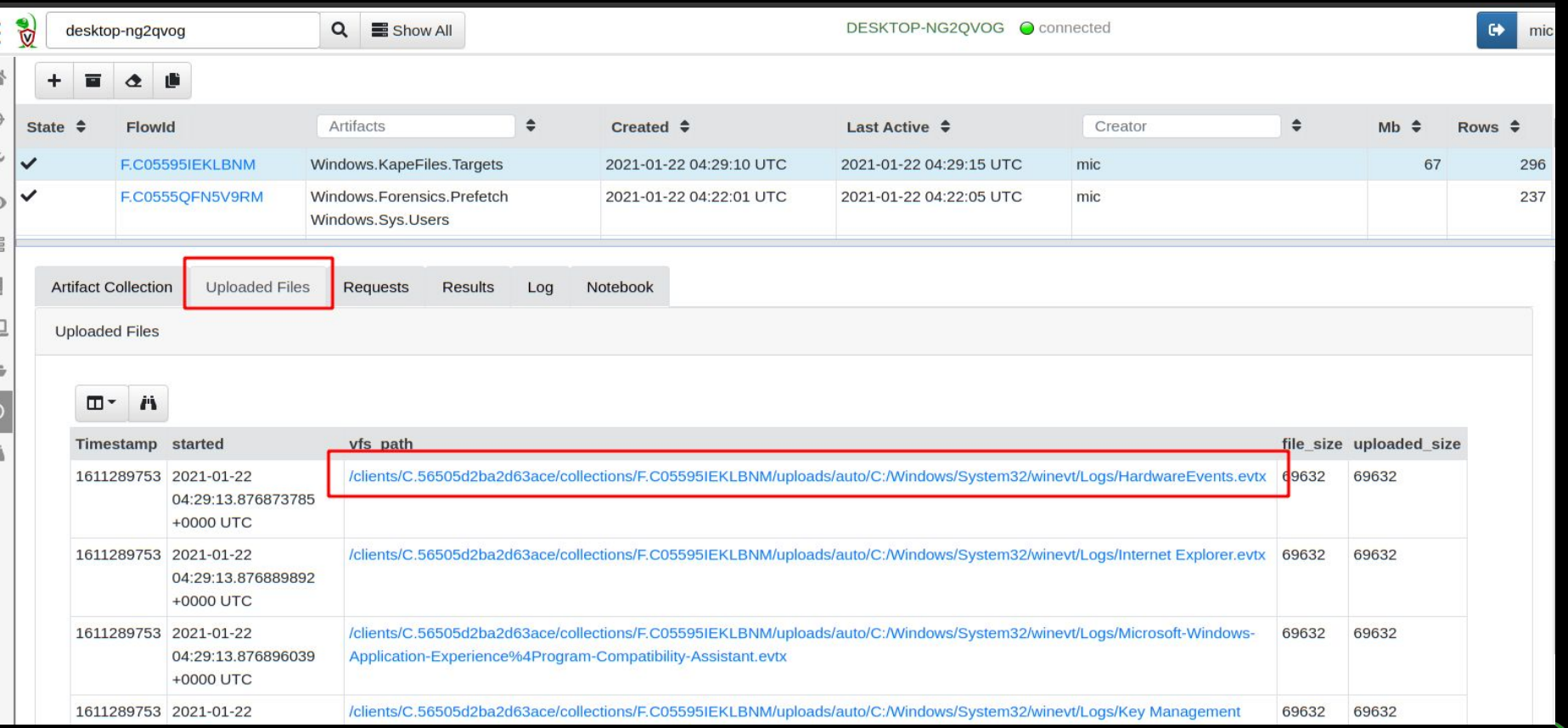


As the query is running on the endpoint any log messages are sent to the server.
Click the log tab to see if there were any errors and how many rows are expected.



Viewing the result tab shows the rows sent from every artifact and source.



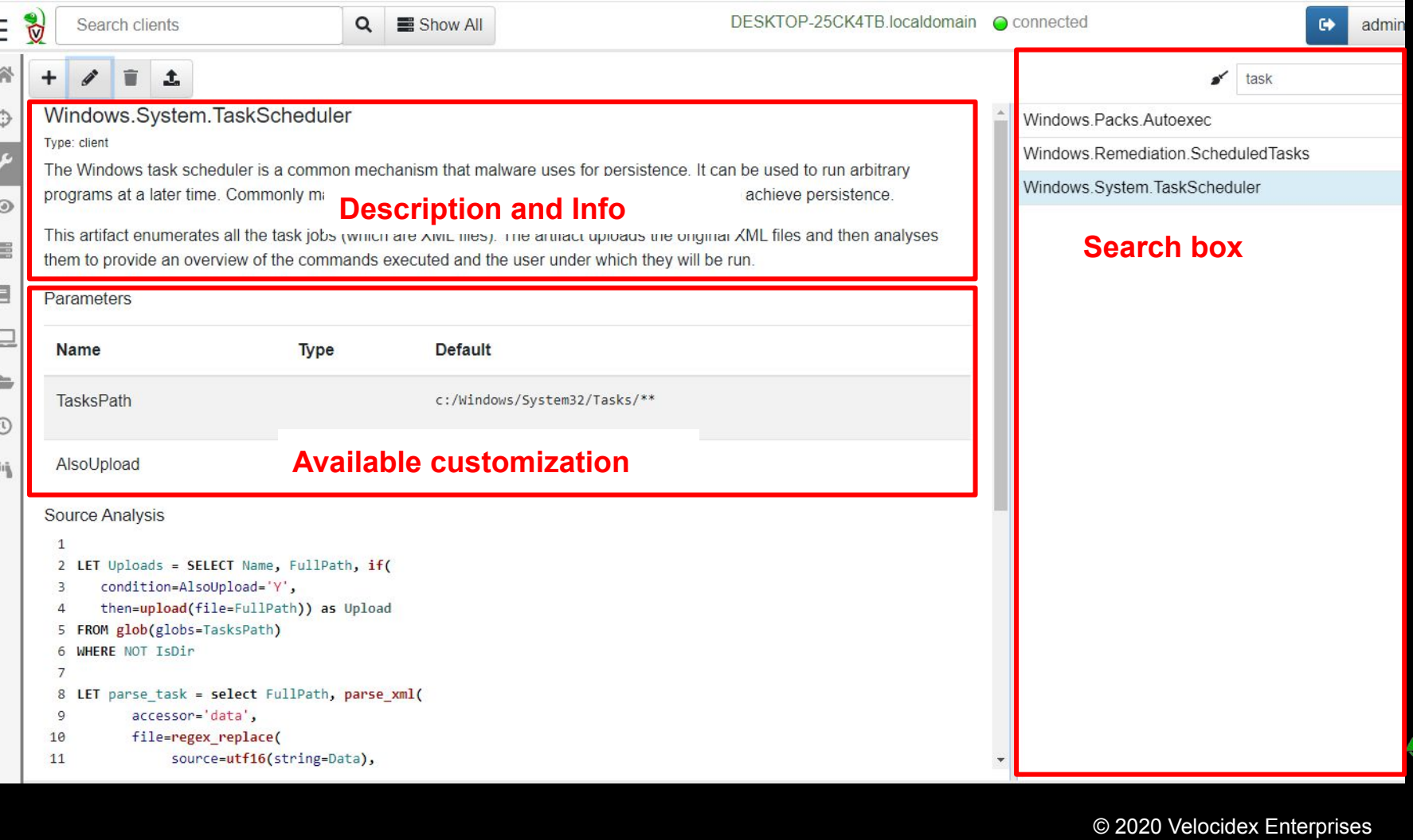


Searching, Viewing and Modifying artifacts

View artifacts

Artifacts are just YAML files
The “View Artifacts” screen allows users to explore the different available artifacts.

While most users will just collect existing ones, we expect power users to customize and write their own artifacts from scratch.



Edit Windows.System.TaskScheduler

```
name: Custom.Windows.System.TaskScheduler
description: |
  The Windows task scheduler is a common mechanism that malware uses
  for persistence. It can be used to run arbitrary programs at a later
  time. Commonly malware installs a scheduled task to run itself
  periodically to achieve persistence.

  This artifact enumerates all the task jobs (which are XML
  files). The artifact uploads the original XML files and then
  analyses them to provide an overview of the commands executed and
  the user under which they will be run.

parameters:
  - name: TasksPath
    default: c:/Windows/System32/Tasks/**
  - name: AlsoUpload
    type: bool

sources:
  - name: Analysis
```

User artifacts must have the prefix “Custom.”. You can collect the original or the customized version as you please.

Close Save

Customizing the dashboard

The main server dashboard is just an artifact called **Server.Monitor.Health** !

You can therefore modify it.

I usually put the name of the deployment prominently and/or links to MSI or client config files - we have so many different deployments it is hard to keep track!

Edit Server.Monitor.Health

```
33      {{ define "CurrentConnections" }}
34        SELECT * FROM sample(
35          n=atoi(string=Sample),
36          query={
37            SELECT _ts as Timestamp,
38               client_comms_current_connections
39            FROM source(source="Prometheus",
40                        artifact="Server.Monitor.Health")
41          })
42      {{ end }}
43
44      ## Server status (My Special Server)
45
46      <p>The following are total across all frontends.</p>
47      <span class="container">
48        <span class="row">
49          <span class="col-sm panel">
50           CPU and Memory Utilization
51           {{ Query "CPU" | LineChart "xaxis_mode" "time" "RSS.yaxis" 2 }}
52          </span>
```

Close Save

The template contains markdown composed from Golang Template Language. You can also run VQL in dashboards!

Hunting everywhere

Hunting

Collecting the same artifact from many endpoints is called “hunting”.

A hunt is just a logical container for many individual collections

- You can download all collections at the same time
- You can see how many endpoints participated
- You can select which machines will participate based on labels, OS or other conditions.

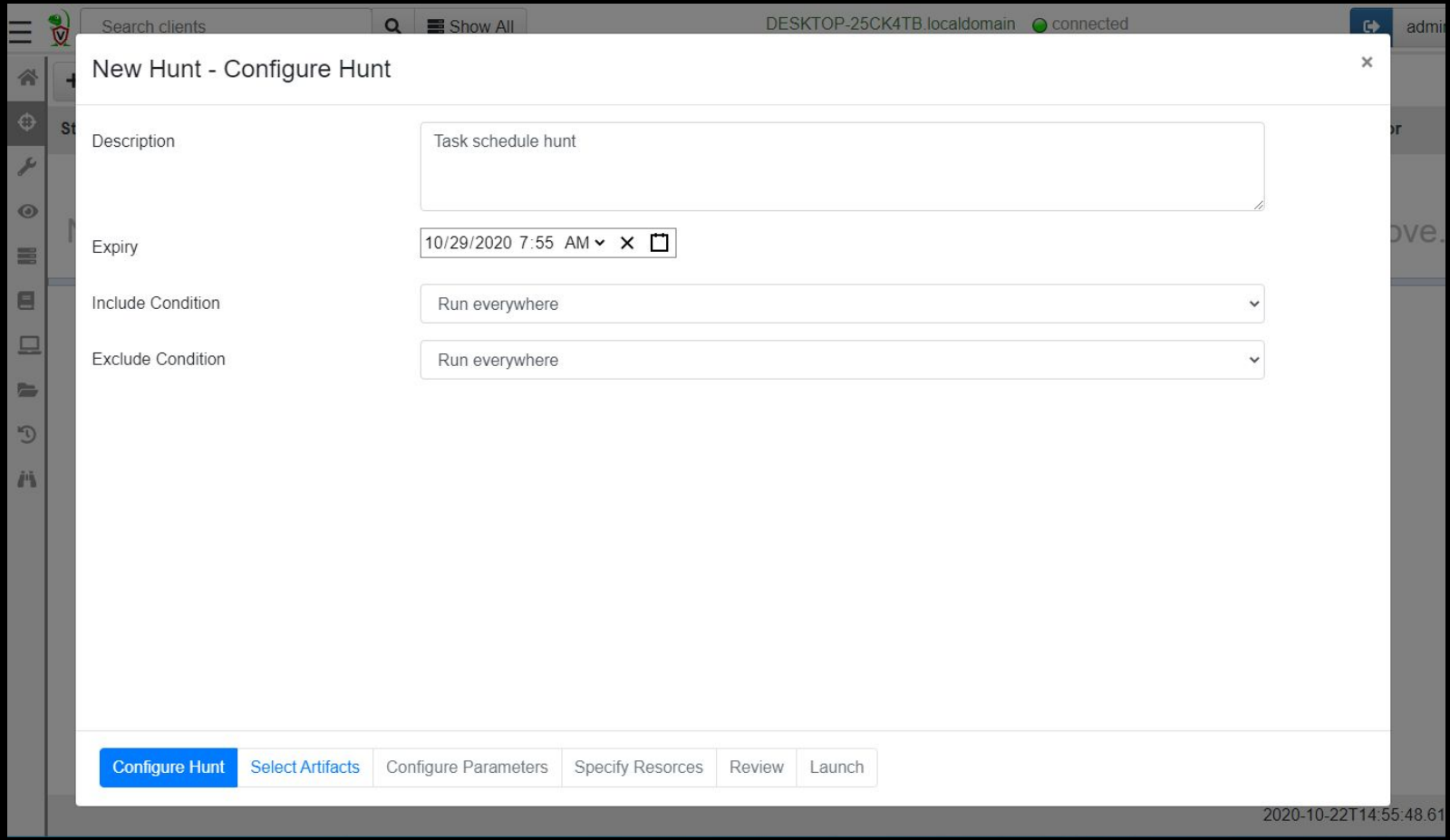
Hunting

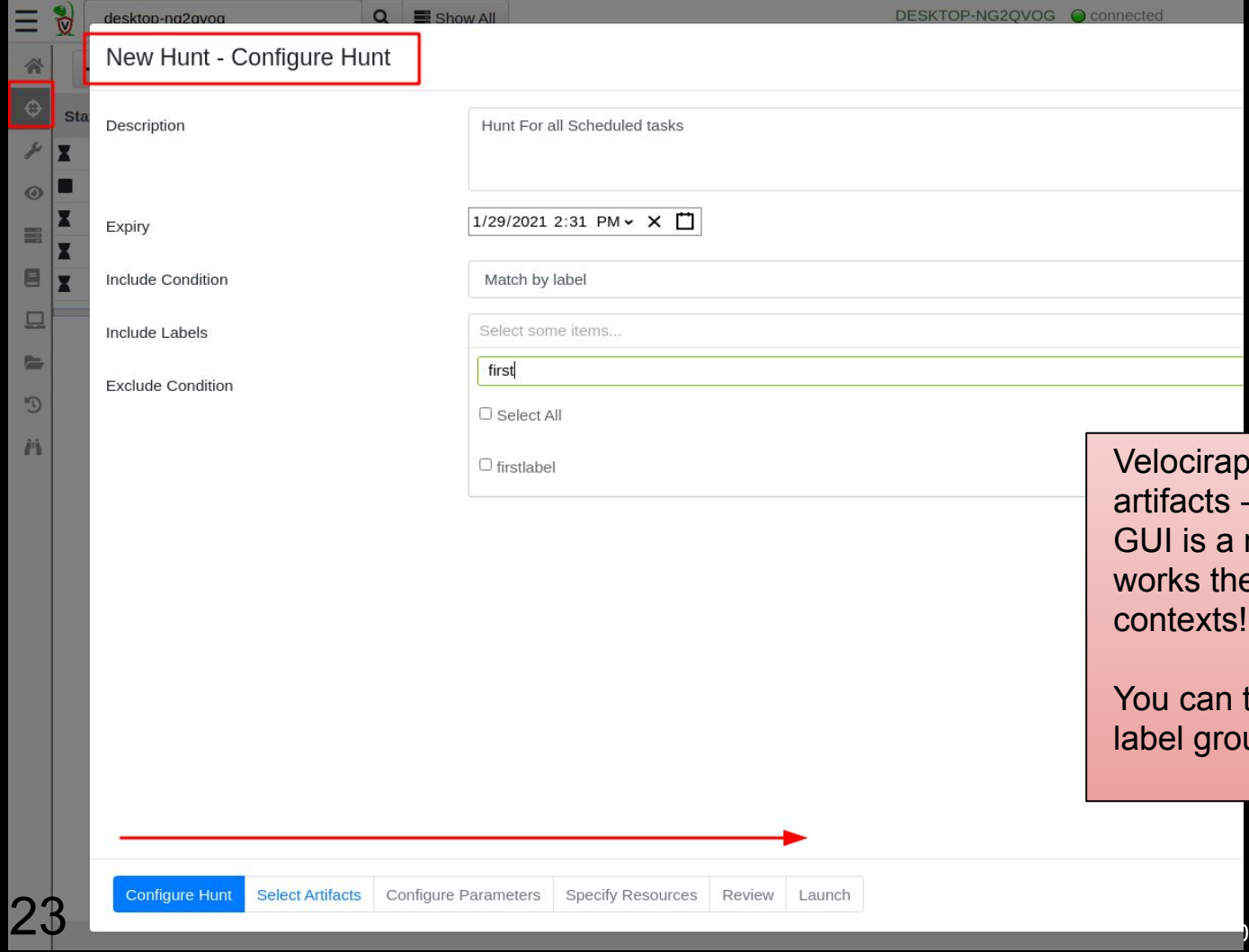
Velociraptor hunts are always active until they expire
Endpoints not currently online will receive the hunt when they check in next.
Therefore the result set is always changing - you can prepare a new download to obtain the latest version of the hunt results.

Exercise - collect tasks everywhere

Repeat the previous artifact collection as a hunt.

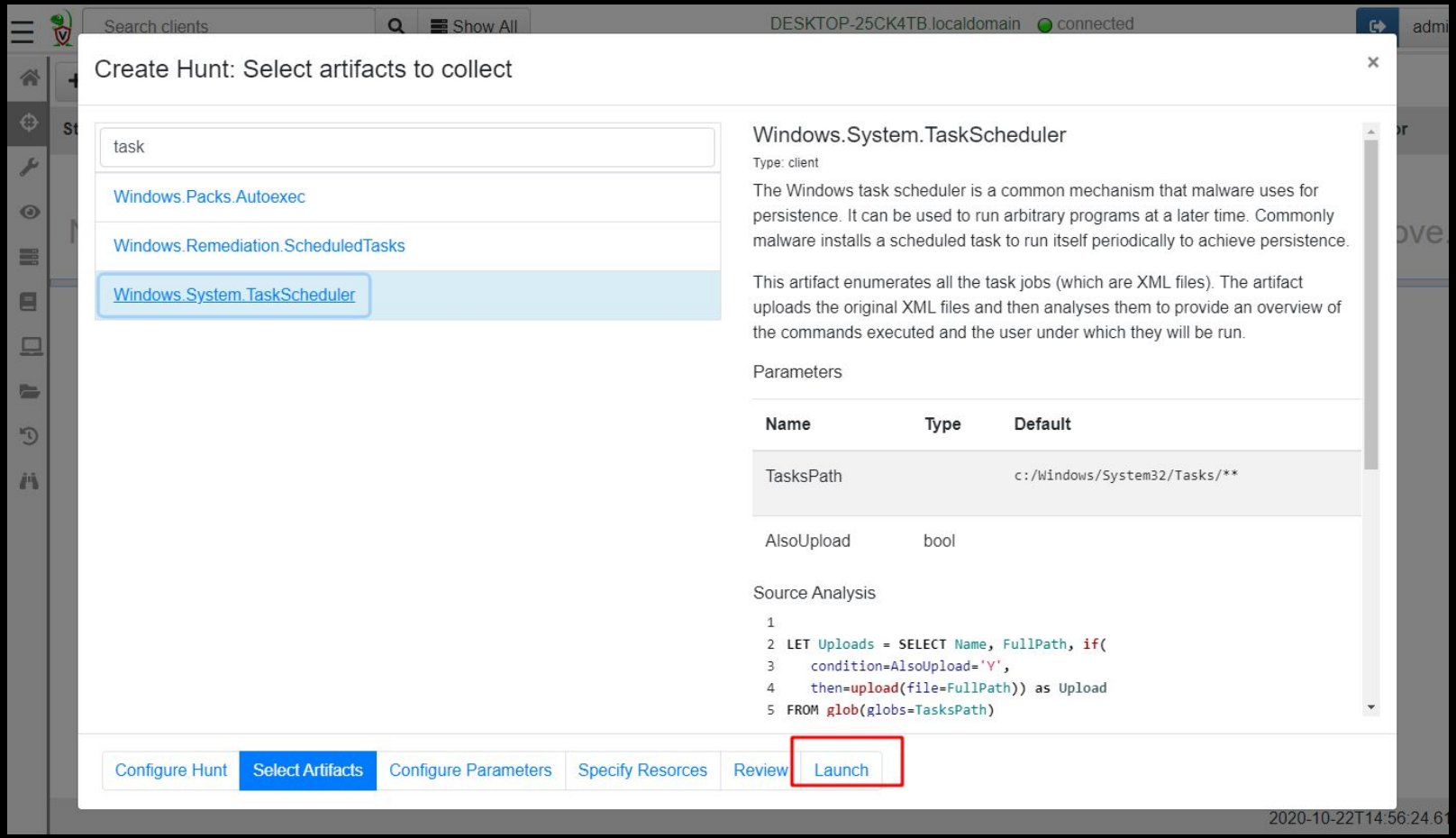
This captures the state of the deployment at a point in time when the hunt was collected.

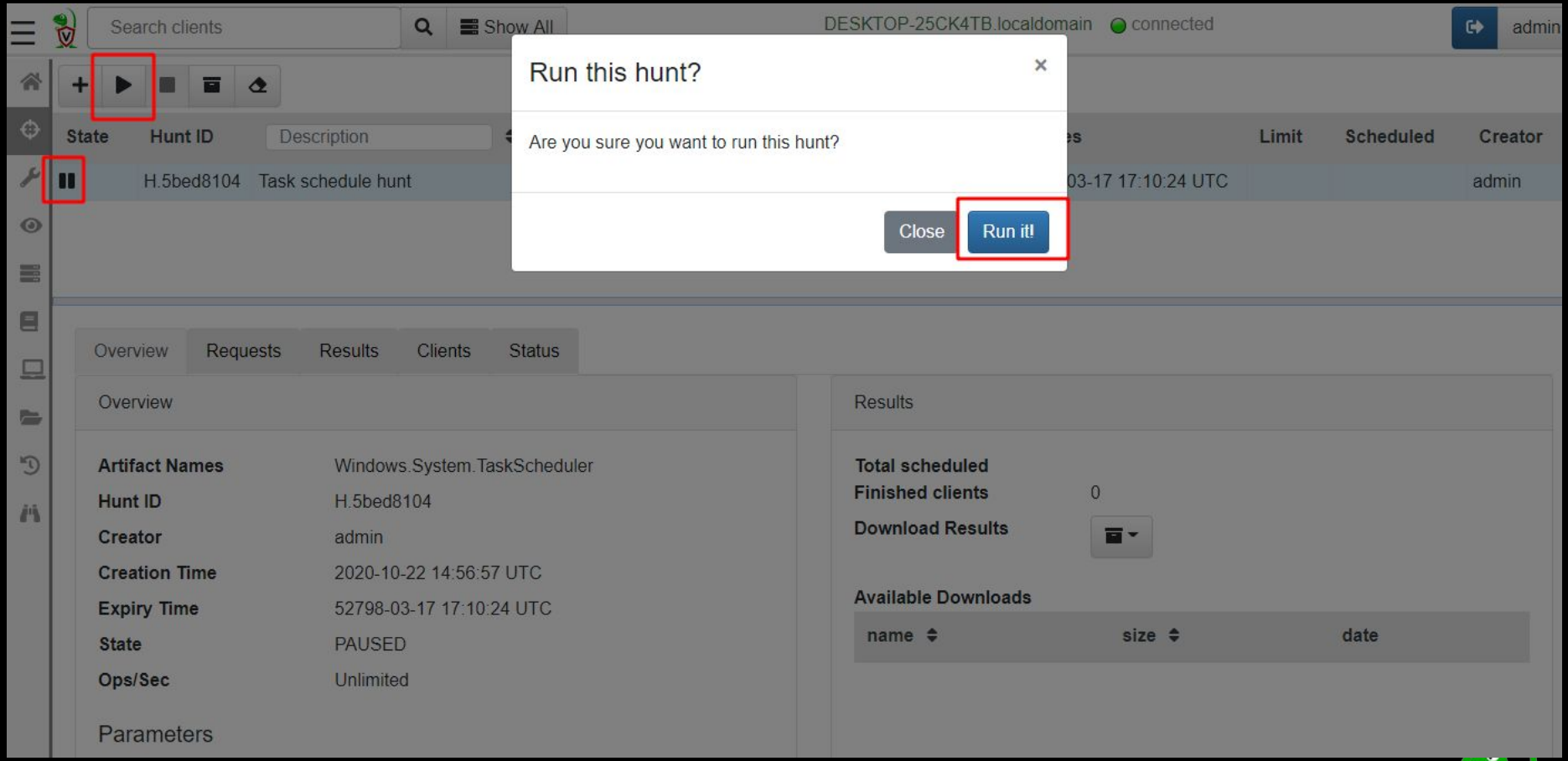


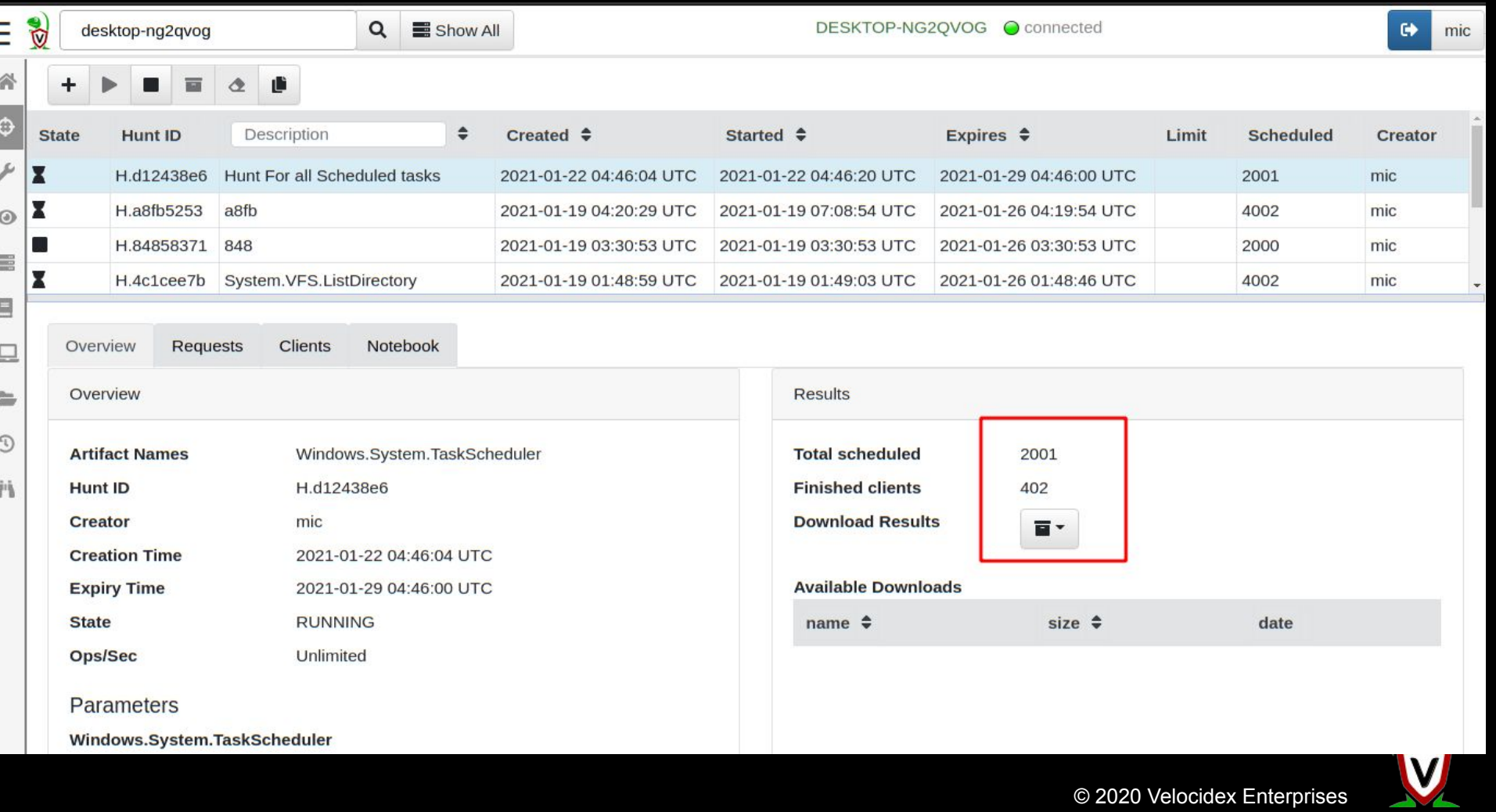


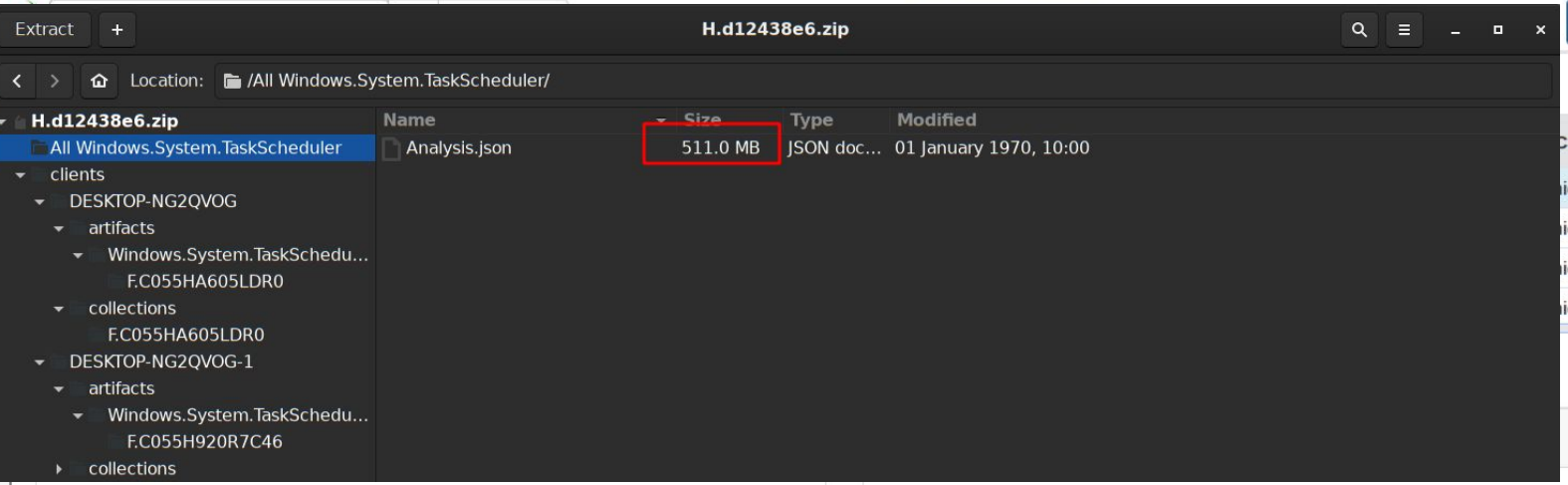
Velociraptor just collects artifacts - the artifact selection GUI is a repeating theme thats works the same way in different contexts!

You can target hunts at specific label groups or OS.

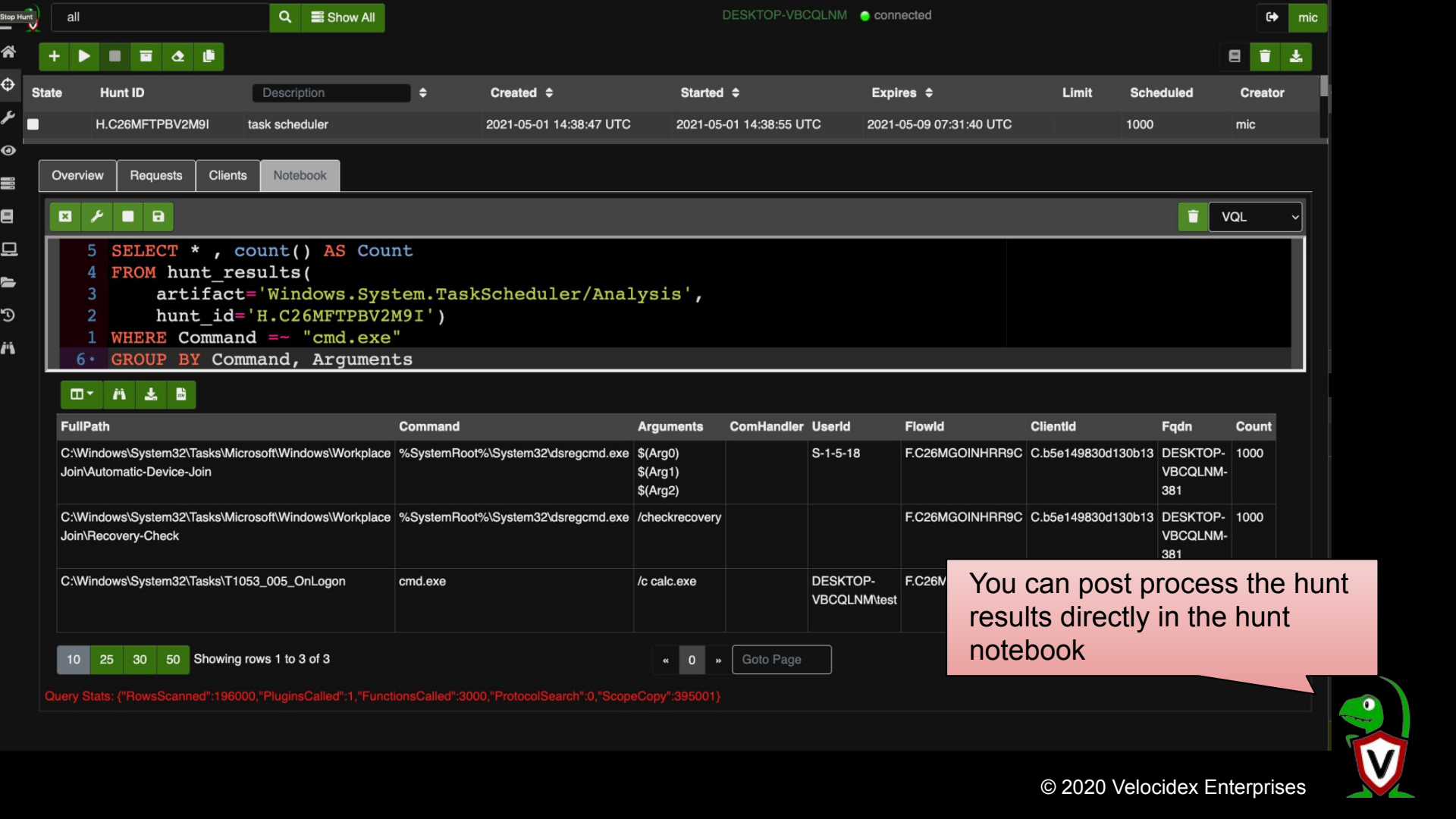








When hunting large numbers of endpoints data grows quickly!



The Velociraptor Reverse Proxy

Velociraptor has a built in reverse proxy

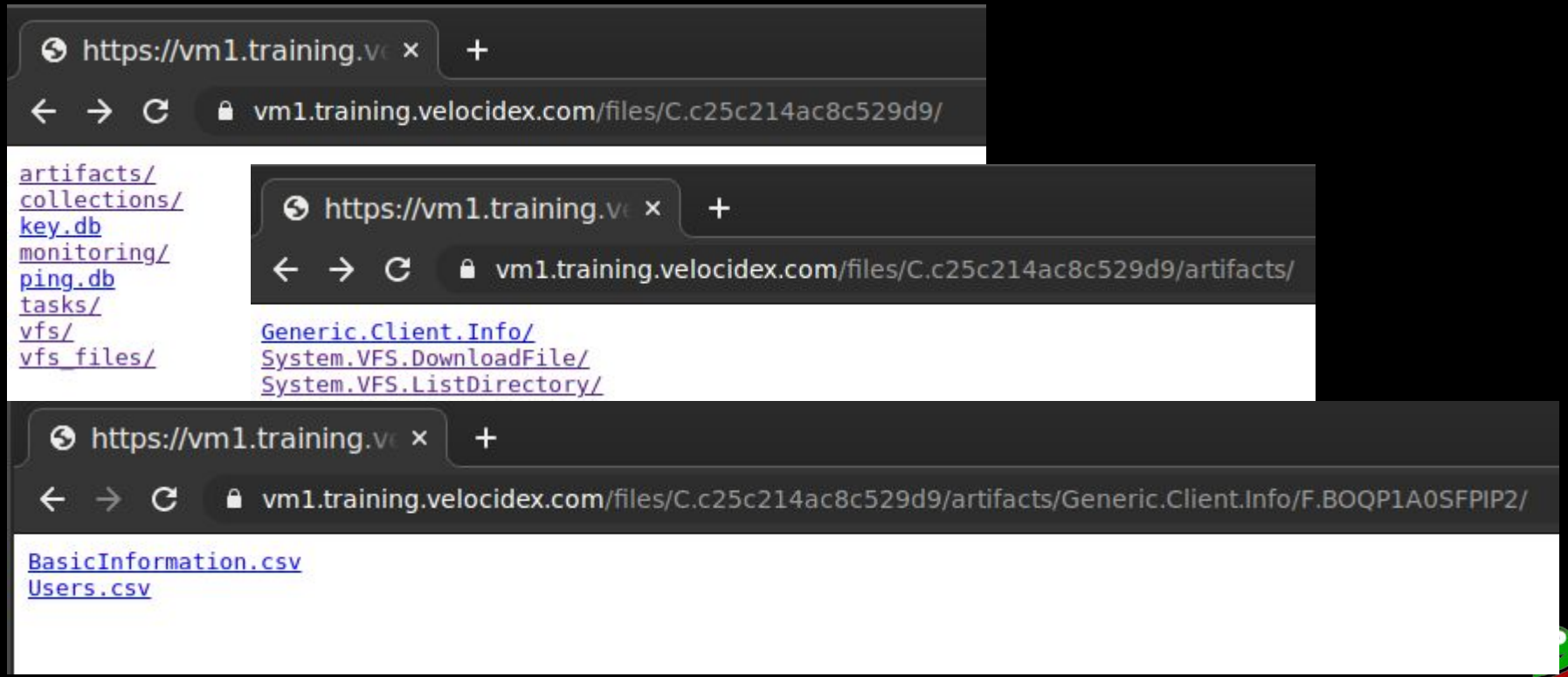
- This allows us to serve other web applications through the Velociraptor server. Velociraptor will take care of authentication and SSL for free.
- It is useful to export the filestore so users can just download the files they want.

Export the file store over HTTPS

```
GUI:
  reverse_proxy:
  - route: /files/
    url: file:///var/tmp/velociraptor/clients/
    require_auth: true
```

```
root@server-1:/home/mike# vi /etc/velociraptor/server.config.yaml
root@server-1:/home/mike# service velociraptor_server restart
root@server-1:/home/mike# service velociraptor_server status
● velociraptor_server.service - Velociraptor linux amd64
   Loaded: loaded (/etc/systemd/system/velociraptor_server.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-02-03 02:05:56 UTC; 938ms ago
 Main PID: 24029 (velociraptor)
    Tasks: 7 (limit: 4915)
   CGroup: /system.slice/velociraptor_server.service
           └─24029 /usr/local/bin/velociraptor --config /etc/velociraptor/server.config.yaml frontend

Feb 03 02:05:56 server-1 systemd[1]: Started Velociraptor linux amd64.
```

Browse the internal file store and note the location of different files.



Double check your security

It is **really** important that auth is required!
Test this twice!
Try to get one of the URLs with no authentication using curl - it should redirect to the auth screen.

```
root@server-1:/home/mike# curl https://vm1.training.velocidex.com/files/C.81bf6660b9db0193/artifacts/
<a href="/auth/google/login">Temporary Redirect</a>.
```


```
root@server-1:/home/mike# curl https://vm1.training.vel
<a href="/auth/google/login">Temporary Redirect</a>.
```


Conclusions

In this module we introduced Velociraptor - a powerful endpoint visibility solution

We mentioned that Velociraptor is based on VQL - a flexible query language

We installed Velociraptor in a cloud deployment, prepared custom MSI packages and distributed them using group policy to our endpoints.

Conclusions

We introduced the Velociraptor GUI

- The Virtual Filesystem abstraction (VFS) provides server side caching of the client's filesystem
  - We can navigate and refresh our view of the client's filesystem in a familiar way.
- We learned about artifacts as a way of encapsulating VQL queries in a human readable, functionally focused YAML file.

Conclusions

- We learned how artifacts can be collected from one end point
  - Exporting the collection into a zip file can archive the files collected and query results as CSV files.
- Leveling up, we can collect the same artifact from many systems. This is called a hunt.
  - Exporting the hunt as a Zip file allows large collections to be archived as a snapshot from the entire deployment.